Detecting Vandalism on Wikipedia across Multiple Languages

Khoi-Nguyen Dao Tran

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

May 2015

Except where otherwise indicated, this thesis is my own original work.

Khoi-Nguyen Dao Tran
May 5, 2015

To my family for your enduring love and support.

To my friends for sharing your life and experiences with me.

To my supervisors for your inspiration, guidance, patience, and feedback.

To the virtual online worlds and gaming communities that are always welcoming with new friendships, experiences, and knowledge.

And to Wikipedia and its supporters for providing the conditions and foundations that have made this thesis possible.

Thank you :-)

Acknowledgments

I thank my family for their enduring love and support throughout my life. There are no words that can express my gratitude for Dat Tran (Dad), Phuong Dao (Mum), Thao Tran (Sister), and our loving dogs Wiggy and Pepper.

I thank my friends for sharing their life, experiences, and time with me; and for helping me through my hardest times while studying for my bachelor and postgraduate degrees at our university. I have spent many hours interacting with friends in online communities and spent countless more with real life friends. Many friends have left parts of their personality with me, and have helped me grow and appreciate life for all its ups and downs. In particular, I thank Lachlan Horne, Jimmy Thomson, and Mayank Daswani for the uncountable number of hours we have spent talking and hanging out, and for the life lessons that I have learnt from each of them.

I thank my main supervisor, Peter Christen, for his inspiration, guidance, patience, and prompt feedback for all aspects of my research. I have been extremely lucky and grateful to have a supervisor who is passionate about his work and the work of his students. I cannot thank him enough for the time he has taken out of his busy schedule to meet, review, and discuss my written work, and to attend my presentations and provide feedback. My research work, skills, and confidence have improved immensely under his supervision.

I also thank my co-supervisors Scott Sanner and Lexing Xie for their continued support and providing guidance when I felt lost in pursuing a solution. I thank them for attending my presentations, and making time for meetings and providing feedback on my written work despite their busy schedule.

In addition, I thank my supervisors, Mamoun Alazab and Roderick Broadhurst, at the cybercrime laboratory for providing me with an opportunity to extend my research work and skills to cybercrime detection, making Chapter 8 possible. The research in Chapter 8 is funded by an ARC Discovery Grant on the Evolution of Cybercrime (DP 1096833), the Australian Institute of Criminology (Grant CRG 13/12-13), and the support of the ANU Research School of Asia and the Pacific. We also thank the Australian Communications and Media Authority (ACMA) and the Computer Emergency Response Team (CERT) Australia for their assistance in the provision of data and support.

I have been part of an amazing research group with many interesting research topics. I thank them for their friendship, encouragements, support, and participation in my presentations and research work.

Finally, I thank the Australian National University (ANU) and the Research School of Computer Science for providing a welcoming environment for study and pursuing research. I have made many friends and have had many interesting, fun, and memorable experiences at the ANU.

Abstract

Vandalism, the malicious modification or editing of articles, is a serious problem for free and open access online encyclopedias such as Wikipedia. Over the 13 year lifetime of Wikipedia, editors have identified and repaired vandalism in 1.6% of more than 500 million revisions of over 9 million English articles, but smaller manually inspected sets of revisions for research show vandalism may appear in 7% to 11% of all revisions of English Wikipedia articles. The persistent threat of vandalism has led to the development of automated programs (bots) and editing assistance programs to help editors detect and repair vandalism. Research into improving vandalism detection through application of machine learning techniques have shown significant improvements to detection rates of a wider variety of vandalism. However, the focus of research is often only on the English Wikipedia, which has led us to develop a novel research area of cross-language vandalism detection (CLVD).

CLVD provides a solution to detecting vandalism across several languages through the development of language-independent machine learning models. These models can identify undetected vandalism cases across languages that may have insufficient identified cases to build learning models. The two main challenges of CLVD are (1) identifying language-independent features of vandalism that are common to multiple languages, and (2) extensibility of vandalism detection models trained in one language to other languages without significant loss in detection rate. In addition, other important challenges of vandalism detection are (3) high detection rate of a variety of known vandalism types, (4) scalability to the size of Wikipedia in the number of revisions, and (5) ability to incorporate and generate multiple types of data that characterise vandalism.

In this thesis, we present our research into CLVD on Wikipedia, where we identify gaps and problems in existing vandalism detection techniques. To begin our thesis, we introduce the problem of vandalism on Wikipedia with motivating examples, and then present a review of the literature. From this review, we identify and address the following research gaps. First, we propose techniques for summarising the user activity of articles and comparing the knowledge coverage of articles across languages. Second, we investigate CLVD using the metadata of article revisions together with article views to learn vandalism models and classify incoming revisions. Third, we propose new text features that are more suitable for CLVD than text features from the literature. Fourth, we propose a novel context-aware vandalism detection technique for sneaky types of vandalism that may not be detectable through constructing features. Finally, to show that our techniques of detecting malicious activities are not limited to Wikipedia, we apply our feature sets to detecting malicious attachments and URLs in spam emails. Overall, our ultimate aim is to build the next generation of vandalism detection bots that can learn and detect vandalism from multiple languages and extend their usefulness to other language editions of Wikipedia.

List of Publications

Below is our list of publications and their relevance to this thesis. We include additional information on the quality of the conferences and journals (or transactions) from their Web sites, email communications, and the Computing Research and Education Association (CORE) of Australasia[1].

CORE provides a ranking of publication venues based on feedback from research peers[2]. The rankings are A* (flagship conference), A (excellent conference), B (good conference), and C (other).

1. **Khoi-Nguyen Tran** and Peter Christen. *Identifying Multilingual Wikipedia Articles based on Cross Language Similarity and Activity*. 2013. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM). CORE ranking of A[3]. Poster paper (acceptance rate of 37.7%).
   This paper describes measures that summarise cross-language knowledge coverage and within-language user activity of Wikipedia articles in different languages and over time. This work is the basis of Chapter 4 with extensions for this thesis of additional measures, visualisations, and analyses.

2. **Khoi-Nguyen Tran** and Peter Christen. *Cross-Language Prediction of Vandalism on Wikipedia Using Article Views and Revisions*. 2013. In Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). CORE ranking of A[4]. Full paper (acceptance rate of 11.3%).
   This paper investigates cross-language vandalism detection using features derived from metadata, and compares performance of multiple classifiers. This work is the basis of Chapter 5.

3. **Khoi-Nguyen Tran** and Peter Christen. *Cross-Language Learning from Bots and Users to detect Vandalism on Wikipedia*. 2015. In IEEE Transactions of Knowledge and Data Engineering (TKDE). CORE ranking of A[5]. Full paper (acceptance rate in the first 10 months of 2014 is 13.2%; impact factor in April 2015 of 1.81).
   This paper investigates cross-language vandalism detection using features derived from text data of Wikipedia articles, and provides an analysis of the contributions of bots in vandalism detection, which is often ignored by related work. This work is the basis of Chapter 6.

[1] http://www.core.edu.au
[2] http://www.core.edu.au/index.php/conference-rankings
[3] http://103.1.187.206/core/25/
[4] http://103.1.187.206/core/1667/
[5] http://103.1.187.206/core/2064/

4. **Khoi-Nguyen Tran**, Peter Christen, Scott Sanner, and Lexing Xie. *Context-Aware Detection of Sneaky Vandalism on Wikipedia across Multiple Languages*. 2015.
   In Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). CORE ranking of A[6]. Awarded "Best Student Paper" at the PAKDD conference. This paper proposes a novel context-aware detection technique for sneaky vandalism, which are more difficult or ambiguous types of vandalism. This work is the basis of Chapter 7.

5. **Khoi-Nguyen Tran**, Mamoun Alazab, and Roderic Broadhurst. *Towards a Feature Rich Model for Predicting Spam Emails containing Malicious Attachments and URLs*. 2013. In Proceedings of the 11th Australasian Data Mining Conference (AusDM). CORE ranking of B[7]. Full paper (acceptance rate of 37%).
   This paper shows the extension of vandalism detection – the detection of malicious activity on Wikipedia – to the detection of malicious content in spam emails from only the email text. This work is the basis of Chapter 8 with extensions for this thesis of a significantly larger data set.

[6] http://103.1.187.206/core/1667/
[7] http://103.1.187.206/core/253/

Glossary

In this section, we list abbreviations and define potentially ambiguous terms in the context of Wikipedia.

en	Language code for English.
de	Language code for German.
es	Language code for Spanish.
fr	Language code for French.
ru	Language code for Russian.
wiki	A Web page or application that allows collaborative editing work to be performed by many people.
Wikimedia	A non-profit organisation seeking to provide free educational content on the Web.
Wikipedia	The world's largest free and open access encyclopedia supported by Wikimedia.
Wikipedias	Plural form means language editions of Wikipedia.
MediaWiki	A free and open source wiki software that Wikipedia uses.
article	An encyclopedic document about a topic.
bot	Bot editor; automated program or software that can edit.
editor	Bot and (registered or anonymous) human editor.
language	Natural or human languages.
page	Another term for an article.
revision	A recorded change to an article that was made by an editor.
user	Human editor; to distinguish from the ambiguous term "editor".
vandalism	Malicious edits, such as changing facts, or inserting obscenities.
AUC-PR	Area under the precision recall curve
AUC-ROC	Area under the receiver operating characteristic curve
CLPD	Cross-Language Plagiarism Detection
CLVD	Cross-Language Vandalism Detection
CRF	Conditional Random Field (a classification algorithm)
MT	Machine Translation
POS	Part-of-Speech
RF	Random Forest (a classification algorithm)
SMT	Statistical Machine Translation
TF-IDF	Term Frequency-Inverse Document Frequency

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wikipedia is the largest free and open access online encyclopedia that attracts tens of thousands of volunteer editors[1] and tens of millions of article views every day[2]. The open nature of Wikipedia facilitates many types of vandals that deliberately make malicious edits such as changing facts, inserting obscenities, or deleting text. Every edit to an article on Wikipedia is recorded as a revision, which means cases of vandalism are seen in the revision history of many articles in many languages. Over the 13 year lifetime of Wikipedia, editors have identified and repaired vandalism in 1.6% of more than 500 million revisions of over 9 million English articles (Kittur et al. [2007] and Section 2.5.2), but smaller manually inspected sets of revisions for research show vandalism may appear in 7% to 11% of all revisions of English articles [Potthast, 2010]. Vandalism survives on the English Wikipedia for an average of 2.1 days and a median of 11.3 minutes [Kittur et al., 2007]. In more recent statistics, vandalism now survives for a median of 75 seconds due to the prevalence and success of counter-vandalism bots [West, 2013].

Vandalism is most persistent, prominent, and well-known to Wikipedia, but vandalism or general malicious behaviour may persist in many other online open collaborative environments (available in many languages) because of their general availability to Internet users. For example, vandalism also occurs in other collaborative sites such as Wiki-based websites [Hasan and Pfaff, 2006], Wikimedia[3] projects, OpenStreetMap (OSM) [Neis et al., 2012], and the Mechanical Turk [Kittur et al., 2008].

Vandalism has not been well studied in these other collaborative environments mainly because of the availability of data for research or the dominance of the (mainly English) Wikipedia projects for Wikimedia in terms of the number of articles, edits, and users[4]. Our reasons for choosing Wikipedia as the main data source are the volume of vandalism data available for many languages; the numerous occurrences of vandalism every day; and the variety of vandalism types [Priedhorsky et al., 2007],

[1] http://stats.wikimedia.org/EN/TablesWikipediansEditsGt5.htm

[2] http://stats.wikimedia.org/EN/TablesPageViewsMonthly.htm

[3] The Wikimedia Foundation is a non-profit organisation that manages many open access projects that aim to allow humans to freely share their knowledge. https://www.wikimedia.org/

[4] The list of Wikimedia projects and their statistics are available at http://meta.wikimedia.org/wiki/List_of_Wikimedia_projects_by_size. Accessed on 21 July 2014.

where each language community has different, but similar definitions of vandalism[5]. Most important for research is, however, the availability and generous licensing of all Wikipedia data[6].

To combat vandalism, editors repair vandalised articles with an edit that removes the vandalised text, or with a revert back to a previous revision. Editors usually leave a comment indicating the occurrence of vandalism. Wikipedia distinguishes many types of vandalism on its policy articles[7] and provides best practice guides to handling vandalism cases. The persistent threat of vandalism has led to the development of automated programs (called bots) and editing assistance programs to help editors detect and repair vandalism [Geiger, 2011], reducing the average exposure time of vandalism and the extra work needed by editors to repair vandalism [Kittur et al., 2007]. Research into improving vandalism detection through application of machine learning techniques has shown significant improvements to detection rates of a wider variety of vandalism. However, the focus of research has mostly been on the English Wikipedia only, which has led us to develop a novel research area of cross-language vandalism detection (CLVD), and evaluate our techniques for five languages: English (en), German (de), Spanish (es), French (fr), and Russian (ru).

CLVD provides a solution to detecting vandalism across several languages through the development of language-independent machine learning models. These models can identify undetected vandalism cases across languages that may have insufficient identified cases to build learning models. We outline two main challenges of CLVD, which have the similar aim of overcoming the language barrier, but from different perspectives of language-independence research (see Section 2.5.1) and cross-domain research (see Section 2.6.1):

1. Language independence: The features developed for machine learning algorithms must be applicable to multiple languages, and have high classification scores within each language. In this thesis, the five languages that we investigate (in Chapters 5 to 7) come from three language families[8]: Romance (Spanish and French), Germanic (English and German), and Slavic (Russian). These European languages share similar text structures that allow language-independent features to be developed. In future work, we aim to investigate how language-independence can be modelled on and between significantly different language families, such as our chosen European languages with Asian (e.g. Chinese, Japanese, and Vietnamese) languages that form some of the largest language communities and articles written on Wikipedia[9].

2. Extendibility: Models of vandalism developed and trained for one language must be generalisable to other languages in the set of working languages.

[5]The types of vandalism can be found for the English Wikipedia at http://en.wikipedia.org/wiki/Wikipedia:Vandalism and other languages from the sidebar.

[6]http://dumps.wikimedia.org/

[7]http://en.wikipedia.org/wiki/Wikipedia:Vandalism

[8]http://en.wikipedia.org/wiki/List_of_language_families

[9]https://meta.wikimedia.org/wiki/List_of_Wikipedias

Importantly, these models must not have statistically significant loss in classification scores when applied across different languages. In this thesis, we address this challenge by exploring the effects of using different classification algorithms in Chapter 5 and different feature sets in Chapters 6 and 7. This ensures vandalism models can be reused and applied to other Wikipedia languages that do not have extensive identification or support for vandalism.

In addition, other important challenges of vandalism detection are:

3. High detection rate: This is a standard challenge to any binary machine learning problem, but in the context of detecting vandalism, the difficulty arises from the ambiguity and evolving nature of vandalism [Potthast, 2010]. Obvious forms of vandalism are easily detectable, but vandals continue to adapt to achieve their own goals. Thus, vandalism models must continually be updated with rare and diverse examples to detect different forms of vandalism. In this thesis, we investigate different features, but in particular, we look to choose the most appropriate classifier for the vandalism task with high detection rate and reasonable trade-offs in model training and testing times.

4. Scalability: Detection techniques must demonstrate their applicability to the large volume of Wikipedia, whether in learning vandalism models or predicting occurrences of vandalism. This ensures timely deployment of new methods, handling of the influx of incoming edits during peak times, and ideally screening for vandalism in real-time as edits are submitted.

5. Variety of data: Wikipedia provides two data types: meta and text, where vandalism may be found through information about the editor, or analysing the text content of a revision. The challenge is to find novel features that can be derived from the meta or text data that allow machine learning algorithms to distinguish vandalism. In this thesis, we focus on developing features to improve classification for one particular machine learning algorithm to avoid the numerous substitutions of different classifiers.

We believe the language barrier is perceived as a limitation to study vandalism in languages other than English. As far as we could find, only one research paper [West and Lee, 2011] addresses vandalism detection in other Wikipedia languages (in addition to English), but its contributions treat each language individually without considering the effects of sharing vandalism models between languages. We show in this thesis that CLVD techniques allow the adaptation of state-of-the-art and future vandalism detection techniques to work across multiple languages while addressing the challenges above.

1.1 Contributions and Thesis Outline

In this thesis, we present our research into CLVD on Wikipedia, where we identify gaps and problems in existing vandalism detection techniques. The following list outlines the aim, motivation, and contributions of each chapter.

- Chapter 2 introduces Wikipedia, the problem of vandalism on Wikipedia and other collaborative environments, examples of vandalism, the counter-vandalism tools, how research is contributing solutions, the general research methodology of CLVD, evaluation measures, and experimental environment. In particular, this chapter highlights two types of data sets used in vandalism detection research and in Chapters 5 to 7: the high quality manually verified (by at least three independent people) vandalism cases from over 62,000 sampled revisions (from three languages) constructed by the PAN workshops in 2010 and 2011 [Potthast, 2010] to encourage development of machine learning techniques; and the vandalism repairs data sets automatically parsed from the Wikipedia data dumps, where we processed over 500 million revisions of over 9 million articles from five different languages in this thesis. This chapter contributes background knowledge for CLVD that is imperative in understanding the vandalism detection research literature and the following chapters of this thesis.

- Chapter 3 provides an extensive survey of research on Wikipedia and vandalism detection research on Wikipedia. We highlight the parts of Wikipedia most relevant for CLVD research, specifically the languages of each article and their associated metadata and text data. In our review of the literature, we begin with a survey of research on the multiple languages of Wikipedia, where the general aim is to summarise, understand, and visualise the volume of knowledge across languages, and the differences in knowledge representation in articles between languages. Then, for vandalism detection, we survey approaches that use two types of data sets: self-constructed by individual researchers, and the PAN Wikipedia Vandalism data sets that provide a reference point for all future vandalism detection research. This chapter contributes a substantial survey of research relating to CLVD and to vandalism detection in general that is not seen in the vandalism detection research literature.

- Chapter 4 proposes measures that summarise the knowledge coverage of Wikipedia articles across languages and user activity on articles within each language. These measures aim to investigate the differences between Wikipedia articles written in multiple languages. We evaluate these measures on over 620,000 Wikipedia articles written in two languages (English and German), present visualisations, and discuss potential recommendation and verification models from these measures for future work. The research contribution of this chapter is a variety of novel information measures to summarise the vast number of Wikipedia articles across different languages. Parts of this chapter have been published in Tran and Christen [2013b] and the remaining parts are extensions for this thesis of additional measures, visualisations, and analyses.

- Chapter 5 investigates using the metadata of Wikipedia articles for CLVD. We demonstrate the use of a new data set for vandalism detection, where a combination with metadata from the revisions improves detection of vandalism.

Importantly, we show that only some classifiers – trained on the metadata features to detect vandalism – are suitable for CLVD as they do not have significant loss in classification performance when classification models are reused across two languages (English and German). The research contribution of this chapter is the additional development (to the vandalism repairs data set described in Chapter 2) of the article views data set for CLVD, where this data set has never before been used for vandalism detection, and improves classification performance when combined with a commonly constructed metadata data set. This chapter has been published in Tran and Christen [2013a].

- Chapter 6 investigates using the text data of Wikipedia articles for CLVD. We propose novel text features for CLVD and also study the often ignored contributions of counter-vandalism bots through these features. We show differences and contrast features important to bots and users in distinguishing vandalism on Wikipedia across five languages. The research contributions of this chapter are the development of novel text features that better distinguish vandalism compared to text features from related work, and an investigation into the contribution of bots to vandalism detection. This chapter has been published in Tran and Christen [2014].

- Chapter 7 develops a novel context-aware vandalism detection method that satisfies the challenges of CLVD by evaluating detection rates across five languages. This method addresses sneaky vandalism – much more difficult or ambiguous types of vandalism – by labelling words and learning dependencies between word labels. We compare the context-aware detection method with the text feature approach of Chapter 6, and analyse the differences in the vandalism detected by each method. The research contribution of this chapter is the development of a novel context-aware CLVD method that detect different types of vandalism compared to text features.

- Chapter 8 extends the text features presented in Chapter 6 to detect malicious content in spam emails. We show that text features for detecting vandalism are also able to detect whether email attachments or URLs are malicious (contains or redirects to viruses). This greatly reduces the need to scan emails as we show email text content is predictive of malicious intent. The research contributions of this chapter are novel text features to detect malicious content in spam and the generalisation of CLVD techniques to other fields. Parts of this chapter have been published in Tran et al. [2013] with extensions for this thesis of a significantly larger data set.

- Chapter 9 concludes this thesis by highlighting significant contributions and interesting research outcomes of the chapters above, and details future research directions of CLVD.

Chapter 2

Background

In this chapter, we present necessary background knowledge for the following chapters of this thesis. We focus only on online environments for occurrences of vandalism. We begin by briefly covering the long and vast history of Wikipedia in Section 2.1, vandalism on Wikipedia in Section 2.2, providing examples of vandalism in Section 2.3, and the response of counter-vandalism on Wikipedia through the development of counter-vandalism tools in Section 2.4. In Section 2.5, we describe in detail the PAN Wikipedia data sets developed and used by research, and self-constructed vandalism data sets often seen in research, including our own constructed in this thesis. In Section 2.6, we describe the general methodologies of vandalism detection and cross-language research used in related work and in this thesis. In Section 2.7, we describe the evaluation measures used throughout this thesis. Finally, in Section 2.8, we give details of our experimental environment, where all experiments in Chapters 4 to 7 were conducted.

2.1 A Brief History of Wikipedia

Wikipedia was first formally released on 15 January 2001 by Jimmy Wales and Larry Sanger[1]. The ideology of a free repository of knowledge that anyone can contribute to was in great contrast to the available encyclopedias at the time that were written by small groups of experts. The popularity of Wikipedia slowly increased until the software underlying Wikipedia underwent massive changes around 2004 to handle the increasingly large user base and number of articles. Afterwards, Jimmy Wales presented his vision to the media, where the public immediately adopted Wikipedia's ideology. This led to tremendous growth of the English Wikipedia in contributing editors and number of articles between 2004 and 2007. The growth was also reflected in the other major (but mainly European) languages of Wikipedia.

Since 2007, the growth of Wikipedia has slowed dramatically across all major languages in the number of new articles created because of the scarcity of easy to write topics [Suh et al., 2009]. The slowing growth is also attributed to some other major problems such as the skewed demographics of Wikipedia editors, the increas-

[1] https://en.wikipedia.org/wiki/History_of_Wikipedia

ingly complex bureaucracy that surrounds many articles, and the harsh stance to new editors by long-time editors [Halfaker et al., 2011]. These have all contributed to the poor retention of new editors and prevention of many infrequent editors from returning. However, in recent years, the Wikipedia communities have recognised these problems and are actively working to create a better image and encourage new people to contribute to Wikipedia [Halfaker et al., 2014].

Currently, Wikipedia is the largest free and open access online encyclopedia that attracts tens of thousands of volunteer contributors and tens of millions of article views every day across over 285 languages. It is the sixth most accessed Web site in the world according to the Alexa rankings[2]. Wikipedia aims to provide a free repository of knowledge for humans, but increasingly, knowledge is also being structured to be used and exploited by computer systems [Sen et al., 2014]. These computer systems, known as bots, that operate internal and external to Wikipedia are the future for Wikipedia as more tasks in writing and maintaining Wikipedia become automated. In this thesis, we aim to develop the research base for new types of counter-vandalism bots to help maintain the high quality of Wikipedia across many languages.

2.2 Vandalism, Online and on Wikipedia

Vandalism in an online context is the malicious modification of text commonly attributed to the wiki collaborative management tool. However, there are a variety of other collaborative software systems that may also suffer from malicious activity akin to vandalism. For example, the Amazon's Mechanical Turk[3] provides a means to outsource human intelligence to tasks that computers currently do not do well, but careful design in experiments or tasks is needed because of inconsistencies with human intelligence or humans exploiting design flaws for monetary gains [Kittur et al., 2008]. OpenStreetMap[4] (OSM) allows people to contribute to providing map details in a similar way to Wikipedia for encyclopedic details, and so this model of open contribution also suffers from vandalism [Neis et al., 2012]. Some examples of vandalism[5] is seen in the OSM wiki[6] with a fake town and graffiti on a map, in the data working group for disputed names[7] for regions of the world such as Jerusalem and Crimea, and also in other languages such as German (seen in the German mailing list[8]) where a fictional island of Lummerland from a children's book was placed on the OSM. These cases show vandalism is an ongoing problem for many of these open collaborative environments (beyond Wikipedia) across many different languages.

The wiki collaborative management tool, named "MediaWiki", is the most widely

[2] http://www.alexa.com/siteinfo/wikipedia.org
[3] https://www.mturk.com/mturk/welcome
[4] http://www.openstreetmap.org
[5] http://forum.openstreetmap.org/viewtopic.php?id=20216
[6] http://wiki.openstreetmap.org/wiki/Vandalism
[7] http://wiki.openstreetmap.org/wiki/Data_working_group/Disputes
[8] https://lists.openstreetmap.org/pipermail/talk-de/2009-June/049150.html

used and well-known software for collaborative work; it is supported by the Wikimedia Foundation and volunteers. Wikimedia has many projects[9] for storing and sharing knowledge across many languages encompassing encyclopedias (Wikipedia), dictionaries (Wiktionary), books (Wikibooks), news (Wikinews), data (Wikidata), and many others. All these projects suffer from vandalism, but identification and research of vandalism is focused on Wikipedia because of the frequency, scale[10], and diversity of types of vandalism [Priedhorsky et al., 2007]. Vandalism research on Wikipedia is appealing because of the availability and generous licensing of all Wikipedia data.

Each edit to Wikipedia is recorded as a revision, where the latest revision of an article is displayed to readers. Cases of vandalism can be seen in the revision history of many articles across many languages. To combat vandalism, editors can revert the latest revision to a previous revision, where they often leave a comment indicating the occurrence of vandalism. Vandalism is often caught and repaired quickly [Viegas et al., 2004; Priedhorsky et al., 2007; Kittur et al., 2007; Geiger and Halfaker, 2013], but the number of cases of vandalism grows in proportion to the fast growth of Wikipedia. Counter-vandalism bots have been developed to partially relieve the burden on editors. Through keyword search of edit comments, bots (bot editors) and users (human editors) identified and repaired vandalism in 1.6% (bots identified 0.54% and users identified 1.06%) of all (500+ million) revisions in the English Wikipedia [Kittur et al., 2007]. Vandalism may appear in 7% to 11% of all revisions from studies manually inspecting a sampled set of revisions [Potthast, 2010]. The differences in the cases of vandalism (approximately 5% to 9%) suggest very difficult or ambiguous forms of vandalism that may require up to 8 rounds of majority consensus from three different annotators in each round [Potthast, 2010].

Vandalism on open wikis such as Wikipedia range from simple and obvious, to complex and well-written, and also automated and coordinated mass attacks. A simple and obvious vandalising of an article is a simple process: (1) pick an article (or a random one[11]), (2) click the edit tab of the article or on any section, (3) insert, delete, or change the content to something obscene or obviously not belonging to the article, (4) then click save page at the bottom of the editing screen. This is a simple method of vandalism, which is often caught and repaired quickly. After repeated abuse, the IP address or username committing vandalism may be blocked[12]. There are a variety of methods to increase the anonymity, longevity, and breadth of vandalism, which is vaguely documented by Wikipedia[12] to avoid promotion of vandalism techniques or creation of new vandals.

Vandals on Wikipedia range from random users that will always exist in any open wiki to organised groups of vandals driven by ideology that may be well funded[13]. The extreme of organised groups of vandals is a recent phenomenon born from the

[9] http://www.wikimedia.org/

[10] The list of Wikimedia projects and their statistics are available at http://meta.wikimedia.org/wiki/List_of_Wikimedia_projects_by_size.

[11] The wiki software that Wikipedia uses helpfully provides a random article link here: http://en.wikipedia.org/wiki/Special:Random.

[12] http://en.wikipedia.org/wiki/Wikipedia:Vandalism

[13] http://en.wikipedia.org/wiki/List_of_Wikipedia_controversies

popularity and adoption of Wikipedia as the de facto source of knowledge by Internet users. The articles on many contentious issues[14] are often vandalised by interest groups and thus these articles have strict editing permissions. The discovery of these groups of vandals comes from investigations by counter-vandals noticing patterns in the styles and target of the vandalism, and tracing vandal IP addresses to known ideological organisations or their subsidiaries[14]. The people who commit vandalism, however, are largely unknown (in their physical form) and currently we are not aware of any completed and published demographic studies on Wikipedia vandals.

The counter-vandals on Wikipedia have a similar wide range of organisation; from volunteer editors guarding the articles they have edited, to organised counter-vandalism groups providing different contributions. These groups may organise around a particular topic[15], develop tools[16], or train and educate new editors to become counter-vandals[17]. The main group on Wikipedia that coordinates counter-vandalism efforts is the Counter-Vandalism Unit (CVU)[18] that was setup in August 2005 (according to the first edit[19]). The CVU provides a place for counter-vandals to coordinate counter-vandalism efforts, build bots, and refine vandalism identification and counter-vandalism techniques.

Vandalism ranges from obvious to ambiguous and subjective interpretations. In an attempt to standardise and categorise vandalism, each Wikipedia language community have created their own list of vandalism types[20]. The lists differ in the types of vandalism that each language community has identified, but in general the categories fit into those defined by Priedhorsky et al. [2007]: "misinformation, mass delete, partial delete, offensive, spam, nonsense, and other", where "other" means (possibly new) types of vandalism behaviour not covered by any defined categories. These "other" type of vandalism generally contains more difficult types of vandalism that requires new detection techniques. Currently, the majority of vandalism is detected and repaired by users, but increasingly, counter-vandalism bots are taking over the responsibility of maintaining quality on Wikipedia. Typical cases of vandalism occurs in the article text, where there are a variety of vandalism types. However, article structures (such as information boxes) and templates can also be vandalised for wider exposure, but these are often more difficult to perform because of editing restrictions on templates.

The occurrence of vandalism varies from the general "background noise" to targeted attacks on certain topics that may be popular in the news media or contentious ideologies. Direct observation of vandal instances as they occur is often not possible, but from observations of repairs, vandalism follows a cycle corresponding to peak

[14] http://en.wikipedia.org/wiki/Wikipedia:Most_vandalised_pages

[15] For example, the WikiProjects often have an implicit goal of identifying and repairing vandalism. http://en.wikipedia.org/wiki/Wikipedia:Database_reports/WikiProjects_by_changes

[16] http://en.wikipedia.org/wiki/Wikipedia:Counter-Vandalism_Unit/Tools

[17] http://en.wikipedia.org/wiki/Wikipedia:CVUA

[18] https://en.wikipedia.org/wiki/Wikipedia:Counter-Vandalism_Unit

[19] http://en.wikipedia.org/w/index.php?title=Wikipedia:Counter-Vandalism_Unit&dir=prev&action=history&tagfilter=

[20] https://en.wikipedia.org/wiki/Wikipedia:Vandalism_types

usage times of the articles in languages belonging to different regions of the world. Some articles about contentious issues are frequently vandalised[21] by users that may come from multiple countries. For example, the Spanish Wikipedia may have multiple peaks of vandalism as Spain or countries from Latin America have their peak usage times.

The English Wikipedia contains an essay on the possible motivation of vandals[22]. However, we could not identify a formal research study into this topic, which may be due to the difficulty of identifying vandals and obtaining their cooperation. The main types of vandals identified are attention-seeking, extremist, emotionally invested, anti-authoritarian, humour, and others such as disagreements with other users, certain articles, the Wikipedia ideology, or political/religious ideals. In recent years, Wikipedia has been targeted by interest groups seeking to bias or alter history or current events. The fame of Wikipedia has also attracted some notable cases of vandalism that have influenced other media such as magazine and television.

2.3 Examples of Vandalism on Wikipedia

To allow the reader to appreciate the widespread effect vandalism can have, we present a set of self-documented (by Wikipedia editors) vandalism cases from the "Notable acts of vandalism" on the "Vandalism on Wikipedia"[23] article. Furthermore, we present an example of vandalism for each data set of the five Wikipedia languages researched in this thesis.

2.3.1 Notable Acts of Vandalism

- **Halle Berry and The Rolling Stone.** A registered user by the name of "Ciii" added false information about actress Halle Berry's new music album (see Figure 2.1). The album never existed nor was the reference link correct. This act of vandalism lasted from the 2nd December 2006 until 25th December 2006, but it was not marked as vandalism by an editor that checked the legitimacy of the album. Meanwhile, the Rolling Stone website reported a false story on this new album[24], which later circulated around other news networks.

  The vandalism edit by user "Ciii" (shown in Figure 2.1) was later modified by a few other editors[25], some of whom seems to have mistaken the vandalism for legitimate information without checking the source[26].

[21] http://en.wikipedia.org/wiki/Wikipedia:Most_vandalised_pages

[22] http://en.wikipedia.org/wiki/Wikipedia:The_motivation_of_a_vandal

[23] http://en.wikipedia.org/wiki/Vandalism_on_Wikipedia#Notable_acts_of_vandalism

[24] The Rolling Stone article is no longer available, but we can access the earliest snapshot on the 14th December 2006, which shows a description highly similar to the vandalism from Wikipedia: http://web.archive.org/web/20061214073317/http://www.rollingstone.com/rockdaily/index.php/2006/12/11/halle-berry-set-to-ruin-reputation-puffy-wants-dancing-singing-boys-and-more/.

[25] A small selection of the revision history of the article "Halle Berry": http://en.wikipedia.org/w/index.php?title=Halle_Berry&offset=20070101000000&limit=100&action=history&tagfilter=.

[26] The last appearance of the album vandalism before it was removed (23 days after the original

- **Stephen Colbert and Elephants.** On the 31st July 2006, Stephen Colbert on "The Colbert Report" (a popular news satire program) encouraged his viewers to vandalise the Wikipedia article on elephants[27]. The motive seems to be his mocking of knowledge by democracy, where if one could convince a large portion of the public to agree with you, then one can record it as knowledge on Wikipedia. As a demonstration of the easiness to commit vandalism on Wikipedia, Colbert proceeded to vandalise the Elephant article live on air (see Figure 2.2). The result was high Internet traffic to the Elephant article, where it and many related articles had to be protected by Wikipedia administrators. Looking at the revision history of the Elephant article[28], the flood of vandalism continued for many days, where the comments indicate repeats of the show spurred new waves of vandalism.

2.3.2 Vandalism in our Data Sets

We choose vandalism examples from each language data set (described in Section 2.5) with some variety to illustrate the diversity of vandalism and potential difficulty in detecting vandalism. Other (interesting and most current) examples of vandalism can also be found on Wikipedia[29][30][31].

- **English.** The example in Figure 2.3 from the English Wikipedia shows a repair of an obvious type of vandalism that uses crass vulgarities to replace the original text. This type of vandalism can be easily detected by counter-vandalism bots because of the out-of-context (to the article) use of vulgarities.
- **German.** From the German Wikipedia, the example in Figure 2.4 shows a repair of a difficult to detect vandalism by text analysis of the article. An image has been replaced with an unrelated image that has a similar file name (shown in bolded text). The difference of one character is not a typographical error, but a vandal attempting to disguise an unrelated uploaded image with a name almost identical to the original image.
- **Spanish.** The Spanish Wikipedia is another edition that frequently uses counter-vandalism bots to repair vandalism as in the example in Figure 2.5. The vandalism is a replacement of text with an insulting description of an artist. This type of vandalism may be difficult to automatically detect because of correct word features, but the text does not fit the context of the paragraph nor article.
- **French.** The example in Figure 2.6 from the French Wikipedia shows a repair of vandalism made by an anonymous user with an IP address of 90.47.194.165.

vandalism): http://en.wikipedia.org/w/index.php?title=Halle_Berry&oldid=96429055#Discography.

[27] http://thecolbertreport.cc.com/videos/z1aahs/the-word---wikiality

[28] http://en.wikipedia.org/w/index.php?title=Elephant&offset=20060802000000&limit=150&action=history

[29] http://en.wikipedia.org/wiki/Wikipedia:Most_vandalized_pages

[30] http://en.wikipedia.org/wiki/Vandalism_on_Wikipedia

[31] http://en.wikipedia.org/wiki/Wikipedia:Vandalism

URL	http://en.wikipedia.org/w/index.php?title=Halle_Berry&oldid=91630568
Vandalised Paragraph	The December issue of Ebony Magazine confirms that Halle Berry is releasing a new ablum from EZ Records entitled Halle. The album is planned to be released on January 9, 2007.

Figure 2.1: Vandalism about Halle Berry.

URL	http://en.wikipedia.org/w/index.php?title=Elephant&oldid=66977359
Vandalised Addition	THE NUMBER OF ELEPHANTS HAS TRIPLED IN THE LAST SIX MONTHS!

Figure 2.2: Stephen Colbert encouraging vandalism on Wikipedia.

URL	https://en.wikipedia.org/w/index.php?title=Autism&diff=529951685&oldid=529951591
Title	Autism
Editor	Dr.K.
Comment	Reverted 1 edit by Stealthf0rce (talk) identified as vandalism to last revision by SandyGeorgia.
Vandalised Paragraph	"Autism" is a type of player from the 1600s ELO of League of Legends. They are some of the most stupid fucking retards you could ever encounter in life. [...]
Repaired Paragraph	"Autism" is a disorder of neural development characterized by impaired social interaction and communication, and by restricted and repetitive behavior. [...]

Figure 2.3: An example case of vandalism on the English Wikipedia.

URL	https://de.wikipedia.org/w/index.php?title=Antike&diff=2466810&oldid=2462703
Title	Antike
Editor	Srittau
Comment	vandalismus-revert
Vandalised Paragraph	[[Bild:Löwentor_**Myken**.jpg | thumb | Das Löwentor von Mykene]]
Repaired Paragraph	[[Bild:Löwentor_**Mykene**.jpg | thumb | Das Löwentor von Mykene]]

Figure 2.4: An example case of vandalism on the German Wikipedia.

This is one of the most difficult types of vandalism to detect automatically because of the minor replacements in words that significantly changes the meaning of the sentence, but do not show word errors, inconsistencies, nor vulgarities.

- **Russian.** Another simple type of vandalism to automatically detect is the use of wrong character encoding when editing different languages. The example in Figure 2.7 from the Russian Wikipedia shows a (presumably) anonymous Polish vandal with an IP address of 83.21.20.7 editing in a non-Cyrillic script to indicate a nationalist view on the Poland article in Russian.

2.4 Wikipedia's Counter-Vandalism Tools

Wikipedia has a variety of counter-vandalism software tools developed by the community that fall into two main categories: automatic detection (bots) and assisting users (editing applications). We briefly describe three notable bots and three notable editing applications that have a clear presence on Wikipedia in repairing vandalism. Other counter-vandalism tools are detailed by the Counter-Vandalism Unit[32], a community project dedicated to combating vandalism by training new counter-vandals and developing counter-vandalism tools.

Table 2.1 shows the number of bots that exist compared to users (before the year 2013) and how many of each have been active in December 2012. We further discuss the significance of bots and why they must be considered in research in the next section. We do not cover editing applications in this thesis because they are not solely focused on detecting vandalism and there are related works that are actively providing solutions for better user interfaces [West et al., 2010b; Halfaker et al., 2014].

- Notable bots:
  - Anti-Vandal Tool[33] is a bot that monitors the feed of all edits on Wikipedia as they occur. Vandalism is detected by matching words in the edit to a list of vandal words used in past vandalism cases.
  - ClueBot[34] was the most active counter-vandal bot from 2007 to 2011. When this bot inspects an edit, a score is determined from a variety of pattern matching heuristics that includes large changes, mass deletes, controversial topics, targeted celebrities, incorrect redirects, vulgar words, minor sneaky changes (explained in Chapter 6), and others that are added as certain types of vandalism are discovered.
  - ClueBot NG[35] is the successor to ClueBot and also the first Wikipedia counter-vandalism bot to use machine learning algorithms to improve detection rate and lower false positives. ClueBot NG uses a combination

[32] https://en.wikipedia.org/wiki/Wikipedia:Counter-Vandalism_Unit
[33] https://en.wikipedia.org/wiki/User:Lupin/Anti-vandal_tool
[34] https://en.wikipedia.org/wiki/User:ClueBot
[35] https://en.wikipedia.org/wiki/User:ClueBot_NG

URL	https://es.wikipedia.org/w/index.php?title=Arte&diff=61972492&oldid=61972471
Title	Arte
Editor	PatruBOT
Vandalised Paragraph	**Ezequiel es un artista feo, pero es mejor que un niño de 3 años, gusta de Ani y** dado que su definición está abierta a múltiples interpretaciones, [...]
Repaired Paragraph	**La noción de arte continúa hoy día sujeta a profundas disputas,** dado que su definición está abierta a múltiples interpretaciones, [...]

Figure 2.5: An example case of vandalism on the Spanish Wikipedia.

URL	https://fr.wikipedia.org/w/index.php?title=Algorithme&diff=79567568&oldid=79566816
Title	Algorithme
Editor	El Caro
Comment	Révocation de vandalisme par 90.47.194.165 ; retour à la version de Lomita
Vandalised Paragraph	Une [[recette de cuisine]] est un **trisomique**. Elle en contient les éléments **autistes**
Repaired Paragraph	Une [[recette de cuisine]] est un **algorithme**. Elle en contient les éléments **constitutifs**

Figure 2.6: An example case of vandalism on the French Wikipedia.

URL	https://ru.wikipedia.org/w/index.php?title=%D0%9F%D0%BE%D0%BB%D1%8C%D1%88%D0%B0&diff=1199673&oldid=1197217
Title	Algorithme
Editor	83.21.20.7
Comment	rev. vandalismul
Vandalised Addition	POLSKA PONAD WSZYSTKO, RUSKIE ŚCiERWA (Песлу блика По льша [...]
Repaired Paragraph	“Польша” (Песлу блика По льша [...]

Figure 2.7: An example case of vandalism on the Russian Wikipedia.

of predefined rules, Bayesian classifiers, and artificial neural networks to generate a vandalism score for a revision that is passed through a threshold calculation and post-processing filters. Known vandalism instances are collected in a data set for the bot to learn models of vandalism. As the data set grows over time and new machine algorithms are added, it is expected that ClueBot NG will be more accurate in distinguishing vandalism. Some weaknesses of ClueBot NG are: no open or peer-reviewed research of the correctness in identifying vandalism, and the discontentment of editors wrongly accused of vandalism; and the focus of development mainly on the English Wikipedia, as seen in Table 2.2.

- Notable editing applications:
  - Huggle[36] is a browser application that allows fast viewing of incoming edits. It allows users to identify vandalism or non-constructive edits, and to quickly revert them.
  - STiki[37] is a cross-platform application for trusted users to detect and revert vandalism and other non-constructive edits. This application was developed from research [West et al., 2010b] and uses a variety of machine learning algorithms to identify potential vandalism for human editors to inspect. Importantly, it allows users to classify an edit in four categories: vandalism, good-faith revert, pass, and innocent, which feeds back into the algorithms to adjust their models.
  - Snuggle[38] is a browser application designed to allow experienced editors to observe the activities of new editors and distinguish vandals and non-vandals. This application was developed from research [Halfaker et al., 2014] to address the decline in retention of new Wikipedia users. The interface provides four categories to classify edits analogous to STiki, but allows viewing of an editor's editing history and personal messaging to provide feedback to (new) users.

2.5 Wikipedia Vandalism Data Sets for Research

Wikipedia provides monthly data dumps of every revision of every article (from over 285 languages) that is publicly viewable on its Web site. Within these revisions, vandalism can be identified from finding repair reverts made by counter-vandalism bots and users, or from manual inspection of content. For research, manually inspected revisions are much more valuable sources of data but significantly more difficult and expensive to obtain, whereas identifying vandalism repairs are relatively simpler and faster but at the cost of accuracy and reliability. Vandalism detection research often

[36] https://en.wikipedia.org/wiki/Wikipedia:Huggle
[37] https://en.wikipedia.org/wiki/Wikipedia:STiki
[38] https://en.wikipedia.org/wiki/Wikipedia:Snuggle

Table 2.1: Number of unique editors (bots and users) in our data sets. An active editor is one that made an edit during December 2012.

Editor	Bots		Users	
Wiki	Total	Active	Total	Active
en	925	(13.1%) 121	31,427,529	(1.4%) 438,629
de	876	(9.3%) 81	6,347,974	(1.0%) 63,960
es	443	(18.1%) 80	5,030,842	(1.6%) 82,330
fr	478	(17.8%) 85	3,557,384	(1.7%) 60,115
ru	323	(27.2%) 88	2,138,513	(3.0%) 63,649

Table 2.2: Count of revisions repaired by bots across languages. Empty cells are zero counts. Only bots that have repaired vandalism across more than one language are shown.

Bot	en	de	es	fr	ru
ClueBot NG	952,610	1,371			
ClueBot	733,423	2,626	9		
VoABot II	112,033	447	1		
DASHBot	9,358	23			
CounterVandalismBot	7,203	21			
AntiVandalBot	2,200	14			
AVBOT	402	51	143,077		
EmausBot	14	5	10	5	4
Salebot		60		47,511	
PatruBOT		6	42,971		
Botarel		2	21,482		

presents results or evaluation using only one of these two data sourcing method. In this thesis, we dedicate two chapters (Chapters 6 and 7) to vandalism research on both data sourcing methods to compare the advantages and disadvantages of each method.

2.5.1 Manually Inspected Data Sets from the PAN Workshops

Identifying vandalism by manual inspection is expensive in people and time, which limits the number of revisions that can be inspected. The interpretation of vandalism differs amongst Wikipedia users, which can lead to incomplete or inconsistent labelling of vandalised revisions on Wikipedia. Potthast [2010] developed two corpora by crowd-sourcing votes on whether a Wikipedia revision contains vandalism using the Amazon's Mechanical Turk[39].

The PAN workshops in 2010 and 2011 held competitions to encourage development of machine learning based vandalism detection methods. The corpus PAN-WVC-10 contains over 32,000 revisions sampled 'important'[40] articles from the En-

[39] https://www.mturk.com/mturk/welcome

[40] 'Important' articles are described without detail as "the average number of times [an article] gets

glish Wikipedia, where 7% of the revisions contain vandalism. The corpus PAN-WVC-11 contains fewer than 10,000 revisions for each of the English, German, and Spanish Wikipedias, where 11% of all revisions contain vandalism.

In these data sets, the lack of detail in the sampling of revisions from 'important' articles may be over-inflating the percentage of vandalism cases. One key disadvantage of the PAN data sets is that they contain very few revisions repaired by counter-vandalism bots, which make them unsuitable for studying the agreement of bots and users in Chapter 6. The PAN-WVC-10 data set contains 14 bots with a total of 101 revisions (0.3% of all revisions), where one bot is a counter-vandalism bot that made a total of 25 revisions (0.07%). The PAN-WVC-11 data set contains a total of 7 bots across three languages, with a total of 34 revisions (0.1%), where one bot is a counter-vandalism bot that made a total of 5 revisions (0.02%). Clearly, we cannot effectively learn and compare bots and users with these few revisions made by bots. Table 2.3 shows the statistics of the PAN data sets with break down by language and known bots.

2.5.2 Vandalism Repairs Data Sets

The repair method of sourcing vandalism is simple, fast, and allows up-to-date vandalism data to be generated for new data dumps. This method (detailed below) employs scanning of the revision comment for indicators of vandalism left by the bot or human editor. We apply this method of generating the vandalism repair data sets used in this thesis. We report our largest and most recent data sets in Table 2.4. See Appendix A for an example of the full processing method with processed data from our English Wikipedia vandalism repairs data set.

We processed all revisions from 2001 to December 31st 2012 (our cut-off date) of the first data dump available in 2013 for these five languages: English (en), German (de), French (fr), Spanish (es), and Russian (ru). We chose these languages because they have some of the highest number of articles on Wikipedia, where four are the United Nations official languages and the most spoken languages in the world.

These data dumps from Wikipedia in total are approximately 106 GB compressed, and decompress to over 15,000 GB. For comparison, the PAN data sets total to approximately 3.2 GB decompressed. We process these data sets (decompressing on-the-fly) in parallel into segments for further parallel processing in feature generation. Our code and data sets are available on request.

The Wikipedia data dumps contain revisions for every article, but we only use the encyclopedic articles (namespace 0) as these articles are the reason people access Wikipedia. Every edit made on an article on Wikipedia generates a new revision with the full content of the article. When vandalism is discovered, it is usually repaired by correcting the vandalised content or by reverting to a past revision, which copies the past revision to become the current revision. In either case, the repaired revision may contain keywords – such as "rvv" (revert due to vandalism), "vandalism",

edited in a given time frame" [Potthast, 2010].

"...rv...vandal...", and analogues in the other languages – in its comment indicating vandalism was detected and repaired (see Appendix A).

In Chapters 5 to 7, we reduce data size by focusing on the metadata or textual data. For the textual data, we conduct further processing of the difference in the content of the flagged revision with the previous revision. We use the Python unified diff[41] algorithm to obtain lines (marked by a full stop or period) unique to each revision and the lines changed.

To distinguish revisions made by bot editors, we obtain lists of bot names for each language from Wikipedia articles and categories maintaining these lists[42]. We split the revisions into those made by bots and those made by users. We do not distinguish edits made by counter-vandalism tools, nor anonymous and registered users, which we leave as future work.

Using this data processing method, we found approximately 1.6% of all revisions from the English encyclopedic articles are identified cases of vandalism, which is consistent with the method and results from Kittur et al. [2007]. Our work focuses on vandalism that triggers a bot or user to repair the revision. We are not interested in all vandalism cases because from visual inspection of some revisions we find that vandalism is sometimes missed and not usually expanded on, which leads to successive revisions containing the same or very similar vandalism. This will likely result in higher classification scores as the true positive class contains repeated samples. Our rationale is to find revisions that trigger counter-vandalism bots and users to interpret as vandalism, and not the successive revisions containing vandalism that may not have been inspected by counter-vandalism bots and users.

2.5.3 Importance of Bots on Wikipedia

Table 2.1 provides a count of the total and active number of bots and users as found in our data sets. Wikipedia defines an active user as one having performed an action in the last 30 days, which we interpret in our data sets as a user having performed an edit in December 2012. In total, we found 2,053 unique bots amongst all bots reported across the five languages. Our visual inspection of bot names shows many bots have worked or are working across different languages, where some have not reported to or have not been identified by that language community on Wikipedia. We also find many bots are reported as active on Wikipedia, but have not made a contribution to any encyclopedic Wikipedia article (namespace 0). Counter-vandalism bots identify the majority of vandalism, but many other bots have some contributions to detecting and repairing vandalism. We identify some prolific counter-vandalism bots working across our set of languages in Table 2.2.

Table 2.4 summarises the number of revisions in our data sets split by editor type and revision type. For learning (in Chapter 6), we further split the data sets into training sets (all revisions before 2012) and testing sets (all revisions in 2012). The testing sets contain between 9-30% of all revisions for each language.

[41] http://docs.python.org/2/library/difflib.html
[42] E.g. https://en.wikipedia.org/wiki/Wikipedia:Bots/Status

Table 2.3: PAN data sets. Number of article revisions in different languages, split by revision type, and bots and users.

Wiki	Type	Normal		Identified Vandals	
	Editor	Bots	Users	Bots	Users
PAN 2010 en	Count	100	29,945	1	2393
	(%)	0.3%	99.7%	0.1%	99.9%
	Total	30,045 (92.6%)		2,394 (7.4%)	
PAN 2011 en	Count	24	8,818	0	1,143
	(%)	0.3%	99.7%	0%	100%
	Total	8,842 (88.5%)		1,143 (11.5%)	
PAN 2011 de	Count	6	9,395	0	589
	(%)	0.1%	99.9%	0%	100%
	Total	9,401 (94.1%)		589 (5.9%)	
PAN 2011 es	Count	4	8,889	0	1081
	(%)	0.1%	99.9%	0%	100%
	Total	8,893 (89.2%)		1,081 (10.8%)	

Table 2.4: Wikipedia data sets. Number of article revisions in different languages, split by revision type, and bots and users.

Wiki	Type	Normal		Vandal Repairs	
	Editor	Bots	Users	Bots	Users
en	Count	23,577,853	293,243,092	1,819,782	3,592,394
	(%)	7.4%	92.6%	33.6%	66.4%
	Total	316,820,945 (98.4%)		5,115,045 (1.6%)	
de	Count	8,274,593	60,564,993	4,754	189,551
	(%)	12.0%	88.0%	2.5%	97.5%
	Total	68,839,586 (99.7%)		194,305 (0.3%)	
es	Count	8,956,251	32,870,538	218,748	128,189
	(%)	21.4%	78.6%	63.1%	36.9%
	Total	41,826,789 (99.2%)		346,937 (0.8%)	
fr	Count	12,885,088	42,524,023	48,101	169,888
	(%)	23.3%	76.7%	22.1%	77.9%
	Total	55,409,111 (99.6%)		217,989 (0.4%)	
ru	Count	6,710,919	26,192,505	182	46,978
	(%)	20.4%	79.6%	0.4%	99.6%
	Total	32,903,424 (99.9%)		47,160 (0.1%)	

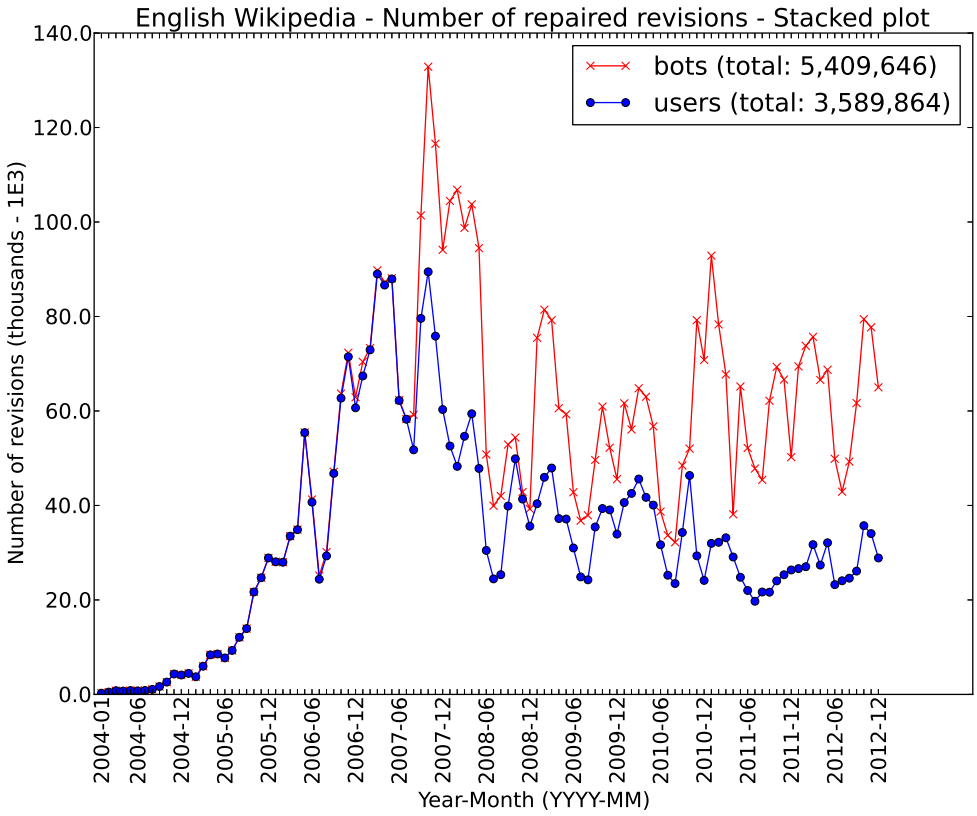

Figure 2.8: Stacked line plot of the number of vandalised revisions identified by bot and users each month in the English Wikipedia.

We show the increasing use of bots to detect vandalism each month in the English Wikipedia in Figure 2.8. In the other Wikipedia languages, we do not see this trend because there may be a bias towards developing bots for the English Wikipedia, a mistrust of bots, or a smaller number of articles for each editor to maintain.

Overall, the normal revisions show bots are actively working in other languages. We see bots sharing a large portion of the workload of over 7% of non-vandalism repair tasks, but with vandalism detection, there is significantly lower usage of bots in non-English Wikipedias. Nevertheless, bots are an important resource for Wikipedia across its languages, and their contributions to vandalism detection cannot be ignored or neglected.

2.6 Research Methodology

The general methodology adopted by this thesis is outlined in Figure 2.9 from an adaptation of Kothari [2004]. Our preliminary study has shown an emerging field of vandalism detection using machine learning techniques to assist editors in maintaining the quality of Wikipedia. After identifying the lack of vandalism detection

in non-English languages, we defined the research problem of cross-language vandalism detection (CLVD), identified the relevant data sets in our literature review, studied the current state-of-the-art techniques, and identified future research directions of the vandalism detection community. We followed these steps in our research:

1. Identify the research problem of vandalism detection in non-English languages.
2. Design new algorithms and techniques for vandalism detection that allows for cross-language research.
3. Develop a theoretical basis for cross-language vandalism detection techniques.
4. Perform preliminary experiments with small sample data sets and prototype algorithms.
5. Design large scale experiments and classification performance evaluation measures to be used.
6. Conduct the experiments.
7. Evaluate the experimental results and compare to the related work in the literature review.
8. Reflect on the results, and if needed make changes to the experimental design and repeat the experiments.

2.6.1 Cross-Language Vandalism Detection Research Methodology

The methodology of cross-language vandalism detection research is an extension to the general research methodology shown in Figure 2.9. We highlight the specific differences in Figure 2.10, which is primarily seen in the experimental design phase, but influences all parts of the general methodology.

The aim of cross-language learning is to overcome the limitation of the small data set size in many Wikipedia languages. Our hypothesis is that using language invariant features we can use large Wikipedia languages to learn and apply vandalism models to smaller Wikipedias without needing to build classification models specifically for those Wikipedias.

Cross-language learning of vandalism means to train a classifier in the training set of one language and apply it to the testing set of another language. This is represented as the connecting lines to the squares of experiments in Figure 2.10. Cross-language learning is a form of transfer learning [Pan and Yang, 2010] which has strong advantages for smaller Wikipedias that do not have the user base to identify and repair vandalism. These few vandalism cases result in low quality vandalism data and a vandalism class imbalance, which are both significant problems in non-English Wikipedias, but they are addressable by extracting appropriate features [Quanz et al., 2012] and feature selection [Wasikowski and Chen, 2010].

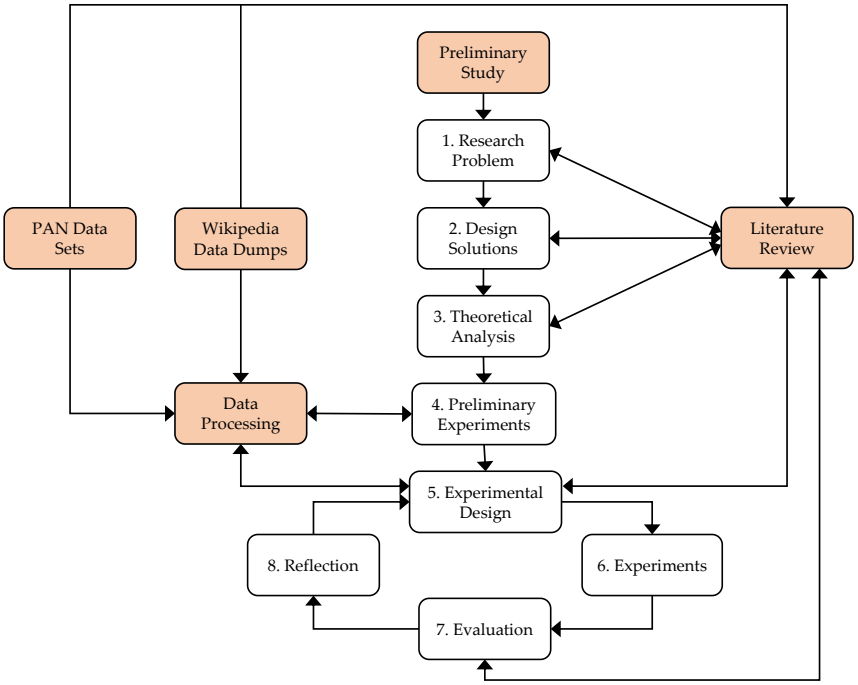

Figure 2.9: General Research Methodology

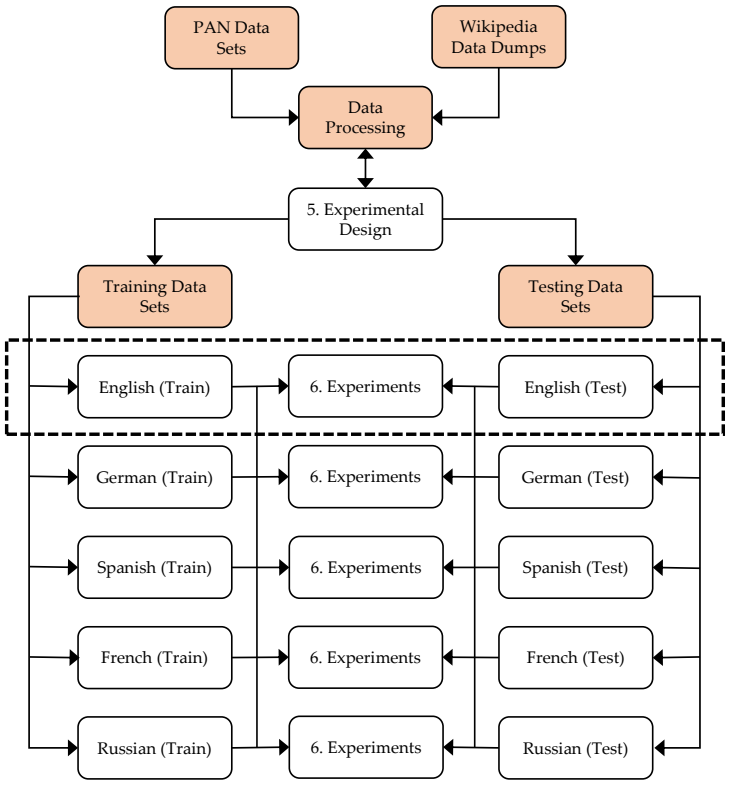

Figure 2.10: Vandalism Detection Research Methodology. The majority of vandalism research focuses on the English Wikipedia as indicated by the dashed box.

The English Wikipedia is the largest Wikipedia, where the majority of vandalism detection research is performed. The methodology of most related work is highlighted as a bold dashed rectangle in Figure 2.10. We demonstrate that cross-language classification is possible without significant loss in classification quality. This allows vandalism detection trained on English to be applied to other languages without needing specific classifiers or additional inputs. Note that we have selected specific features to distinguish vandalism to maximise the classification scores, while avoiding problems of specific cultural knowledge of each language.

2.7 Evaluation Measures

In Chapters 5 to 8, we use the evaluation measures of the area under the precision-recall curve (AUC-PR) and the area under the receiver operating characteristic curve (AUC-ROC) to evaluate our classification tasks. The AUC-PR score gives the probability that a classifier will correctly identify a randomly selected positive sample (e.g. vandalism) as being positive. The AUC-ROC score gives the probability that a classifier will correctly identify a randomly selected (positive or negative) sample. Both scores range from 0 to 1, where a score of 1 means 100% or complete correctness in labelling all samples considered by the measures.

We describe our measures here in brief following Davis and Goadrich [2006], where they also show important relationships between ROC and PR curves. Given an error matrix:

	actual vandalism	actual normal
predicted vandalism	TP	FP
predicted normal	FN	TN

where TP are true positives (correctly predicted vandalism edits), TN are true negatives (correctly predicted normal edits), FP are false positives (incorrectly predicted as vandalism edits), and FN are false negatives (incorrectly predicted as normal edits). Ideally, both FP and FN errors need to be reduced where possible, because FP errors adversely affects new user retention [Halfaker et al., 2011] and FN errors are the more difficult types of vandalism to detect. The precision (P), true positive rate (TPR), and false positive rate (FPR) are defined as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\,, \quad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\,, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$$

The ROC curve is obtained by plotting TPR against FPR, while the PR curve is obtained by plotting P against TPR. In our results, we plot AUC-PR against AUC-ROC to show the trade-offs between P, TPR, and FPR.

AUC-PR is an alternative measure to the AUC-ROC that is often used to evaluate binary classification problems [Davis and Goadrich, 2006]. Davis and Goadrich [2006] demonstrate that a binary classifier with a curve that shows strong performance in

AUC-PR scores will also show strong performance in AUC-ROC scores, but not vice versa. This is evident in related work that promotes strong performance in AUC-ROC scores, but have poor AUC-PR scores (as we show in Section 6.6). This shows the effects of unbalanced classification classes not being considered. Our classification results are for balanced classification classes, but we demonstrate in Section 6.5.5 that AUC-PR scores do not decrease significantly for unbalanced classes.

We use both measures throughout Chapters 5 to 8 (but favouring AUC-PR where results are too numerous) as they address two concerns in vandalism detection: (1) correct prediction of positive vandalism cases (high precision) and finding all positive cases (high recall) (AUC-PR), and (2) the binary classifier needs to distinguish both classes equally well (AUC-ROC). For (1), we are not concerned about the negative (non-vandalism) predictions as they are not detrimental to maintain quality of Wikipedia articles. However, we do not want to be overwhelmed by incorrect negative predictions, which is addressed by (2). We also prefer to report AUC-PR scores over AUC-ROC scores because AUC-PR scores are not influenced by the data sampling in the evaluation Davis and Goadrich [2006].

2.8 Experimental Environment

For Chapters 4 to 7, our relatively small and private server provided an environment for data storage, data processing, performing experiments, and evaluating and analysing experimental results. The server consisted of:

- Intel(R) Xeon(R) CPU E5645 @ 2.40GHz; 12 physical cores, 24 virtual cores.
- 128 GB RAM; 64 GB direct memory access via /dev/shm/ (used as RAM disk).
- 200 GB solid state hard drive, two 2 TB 7200 rpm mechanical hard drives.
- Ubuntu 12.04 LTS (later upgraded to 14.04 LTS) server edition.

Other important software packages for experimental code are:

- Python 2.7.6
  - matplotlib 1.3.1
  - numpy 1.8.1
  - scikit-learn 0.14.1
  - scipy 0.13.3
- bash 4.3.11(1)-release
- screen 4.01.00devel
- xargs 4.4.2

We have only listed information we considered important to replicating our experimental code. There are many other packages and software on various other operating systems that we have used for collating, analysing, plotting, writing, and displaying results in our written work.

2.9 Summary

In this chapter, we presented a background for the contributions in this thesis. We briefly covered the history of Wikipedia, vandalism on Wikipedia, examples of vandalism, counter-vandalism tools, vandalism data sets for research, research methodologies, and our environment for experiments. This chapter lays the foundation for all following chapters.

In the next chapter, we survey the related work on vandalism detection and other similar research on Wikipedia. We cover a variety of research on Wikipedia on multilingual topics, vandalism characteristics, counter-vandalism tools, vandalism data sets for research, and the emerging context-aware vandalism detection methods. These past vandalism research papers have focused almost entirely on the English Wikipedia and on developing features for machine learning. We use these past research papers as inspiration and guidance for the novel contributions of this thesis in the later chapters.

Chapter 3

Related Work

In this chapter, we survey research papers on different aspects of multilingual research and vandalism detection on Wikipedia. These research papers form the basis of our research in Chapters 4 to 8. However, Chapter 8 contains a related work section because many of its relevant related research papers belong to a different domain of research, and so are not included in this chapter.

In Section 3.1, we cover multilingual research on Wikipedia in the topics of machine translation, sentence similarity, information boxes, semantic convergence and coverage of articles, plagiarism detection across languages, and transfer learning. The remaining sections cover vandalism research on Wikipedia. Section 3.2 surveys research that characterises vandalism for analysis and understanding. Section 3.3 identifies research on counter-vandalism tools and the research backed tools. Section 3.4 provides the research resulting from the two main types of data sets covered in Section 2.5. Section 3.5 covers research on past context-aware vandalism detection techniques that specifically relates to Chapter 7. Finally, Section 3.6 concludes this chapter.

3.1 Multilingual Research on Wikipedia

In this section, we highlight multilingual research that specifically relates to Chapter 4, but forms a basis for Chapters 5 to 7. The research papers presented in this section show how to overcome the knowledge that is locked away – wholly and partially – in each language edition of Wikipedia. This thesis extends on these ideas by presenting measures that determine semantic coverage and convergence of articles across multiple languages.

3.1.1 Machine Translation

Machine translation (MT) is a field of research that develops techniques to automatically translate text or speech from one human language to another. Costa-Jussa and Farrus [2014] survey a variety of machine translation techniques, where one of the most popular technique is statistical machine translation (SMT). SMT uses a corpus of parallel sentences (sentences that have the same or similar meaning) in two languages to extract statistical models of words and sentence structures for translations

(see Section 3.1.2 below). In this thesis, we use a complete open source SMT system called Moses[1] [Koehn et al., 2007] to translate Wikipedia articles for summarisation measures to estimate knowledge coverage between languages in Chapter 4. Moses uses the dominant paradigm of phrase-based SMT that has emerged from past MT research [Koehn et al., 2007]. Phrased-based SMT models generally rely on learning translation tables from words and reordering models from context derived from parallel sentences; this is the standard technique used by commercial SMT systems such as Google Translate[2] and Bing Translator[3] [Koehn, 2009].

Koehn [2009] describes other core MT techniques: word-based models, where the frequency of a word translation determines the probable translation, and the alignment probabilities of words determine their lexical ordering in a sentence; decoding, where the translation with the highest probability is determined; and language models, where the quality of translation is determined by evaluating translated text against language models that dictate qualities of a good sentence.

Costa-Jussa and Farrus [2014] survey SMT research by breaking down the challenges of SMT at different linguistic levels for evaluation and identification of potential areas for further research. The linguistic levels are categorised as

- orthography, where the challenges are to determine the correct spelling, truecasing or capitalisation, normalisation, tokenisation, and transliteration of sentences in a language;
- morphology, where the challenge is identify the correct word structure such as conjugations and inflections;
- lexis, where the challenges are to determine the words and phrases that are particular in a language, and how to translate the resulting unknown or uncommon words;
- syntax, where the challenge is to determine the rules of constructing a sentence or determining word ordering in a sentence; and
- semantics, where the challenges are to determine the meaning of words and to choose sensible words for a sentence.

Overall, MT is a complex research field as shown by Costa-Jussa and Farrus [2014], which is beyond the scope of this thesis. We avoid using MT in vandalism detection tasks because of the complexities in developing a translator capable of transferring the information of vandalism across languages. Instead, we develop cross-language techniques that avoid translation by reusing classification models across data sets from different languages.

3.1.2 Sentence Similarity

Sentence similarity is a research area that looks at identifying highly similar sentences between two articles in different languages. To find similar sentences across multiple

[1] http://www.statmt.org/moses/
[2] http://translate.google.com/about/intl/en_ALL/
[3] http://www.microsoft.com/translator/automatic-translation.aspx

languages, research papers focus on specific features of sentences in articles written in many languages. For example, nouns, verbs, and adjectives are possible markers for similar sentences as they represent distinct concepts between languages [Gomaa and Fahmy, 2013]. For Wikipedia, similar sentences can be found by machine translation and the cross-language link structure in articles [Adafre and de Rijke, 2006]. These similar sentences can improve statistical machine translators by providing parallel sentences [Smith et al., 2010], and identify gaps of knowledge for Wikipedia editors [Filatova, 2009].

Other research on sentence similarity on Wikipedia looks to extract sentences to improve machine translation of text. Adafre and de Rijke [2006] propose two paragraph similarity measures using machine translation of text and using the cross-link structure of articles to create a bilingual lexicon. The bilingual lexicon produces fewer incorrect pairs in the overlapping words of translated paragraphs. Smith et al. [2010] present novel methods for extracting parallel sentences from Wikipedia to improve the quality of statistical machine translation. Wikipedia provides a viable source of parallel sentences, as they are often found in close proximity to each other in article pairs. Similar to the two works above, Mohammadi and Ghasem-Aghaee [2010] present an aligned parallel corpus constructed by extracting parallel sentences from Wikipedia using candidate pairs generated from a link based bilingual lexicon. Yeung et al. [2011] look at identifying gaps in alignments of sentences from multilingual Wikipedia articles. The authors develop a system to assist cross-lingual Wikipedia editors to transfer information from other languages to fill these gaps.

For general similarity approaches, Gomaa and Fahmy [2013] survey a variety of text similarity approaches that are extensions for comparing multilingual text. These approaches are grouped into three categories: string-based, where the similarity (or distance) between two strings are determined using approximate string matching [Christen, 2012]; corpus-based, where the similarity between words are determined from large corpora of words such as written or spoken texts; and knowledge-based, where the similarity of words is derived from a database of lexical or semantic relations, such as WordNet[4].

3.1.3 Information Boxes

Information boxes[5] on Wikipedia are a summary of common aspects that certain articles share. For example, countries share common information of flag, capital city, languages, government, population, currency, and so on. These structured parts of a Wikipedia article are similar across languages, which allow fact completion, where parts of information boxes across languages are aligned and missing facts are resolved when possible. Adar et al. [2009] and Bouma et al. [2009] both look at methods of aligning info boxes between two languages and inserting missing data using attributes from the other language. Various similarity measures are used to automatically match attributes. Kulkarni et al. [2012] use Wikipedia info boxes from

[4] http://wordnet.princeton.edu/
[5] https://en.wikipedia.org/wiki/Help:Infobox

multiple languages to automatically determine whether articles across languages are similar at a higher level of abstraction by using homophones and synonyms.

These boxes allow important information to be extracted from Wikipedia for other uses, such as DBpedia[6] [Lehmann et al., 2014], where over 1.46 billion facts that describe over 13.7 million things from 111 language editions were extracted. These facts and things can be used to link other Web data, or queried to discover interesting relations or uses. DBpedia represents a new use of Wikipedia, where more information in articles are becoming structured to tailor for bots from the Internet to query and use Wikipedia as a resource for other Web sites [Sen et al., 2014].

3.1.4 Semantic Convergence and Coverage of Articles across Languages

This section is mainly for Chapter 4, where we propose new measures of semantic convergence of Wikipedia articles across languages. Chapter 4 focuses on measures for the semantic coverage and the semantic convergence of articles in different languages. Our measures can assist editors to identify articles that have semantically converged in one language, and allow them to improve the semantic coverage.

The semantic convergence of articles across languages can be determined by evaluating the similarity of sequential revisions of Wikipedia articles. When the similarity of content between revisions falls below a threshold, continuing edits do not change the meaning of articles [Thomas and Sheth, 2007]. Following edits do not change the meaning of articles, and these articles can be considered to be mature. Thomas and Sheth [2007] use the term frequency-inverse document frequency (TF-IDF) [Manning et al., 2008] representation of articles in their similarity measure. This is a distinction from other research on similarity and Wikipedia because TF-IDF gives higher weights to the important words in articles, and meanings are likely to be derived from these important words.

The semantic coverage of Wikipedia articles across different languages shows how information is shared across languages. Filatova [2009] looks at the semantic coverage of Wikipedia biographical articles in different languages by identifying overlapping descriptions between different languages. The identified missing descriptions or facts between languages can be resolved by translation. Yeung et al. [2011] develop a system to assist multilingual Wikipedia editors to transfer information between languages. The system works by identifying what information is missing and the probable location to insert the information from different languages, which have been translated accordingly to the language of the targeted article.

For an overview of the semantic coverage of Wikipedia articles, a system named Omnipedia [Hecht and Gergle, 2010; Bao et al., 2012] shows how information is shared and the uniqueness of information across 25 languages. Omnipedia provides a user interface that allows users to discover similarities and differences in articles across languages. Bao et al. [2012] describe how similarities, differences, information diversity, and missing content in articles written in different languages are much

[6] http://dbpedia.org/About

greater than previously assumed. To facilitate this user interface, some of the underlying algorithmic challenges of Omnipedia are alignment of article content, links, and ambiguities across languages. A study of Omnipedia is presented, which looks at how people interact with multilingual information. The study contains 27 participants, with one user notably remarking: "It's ridiculous how many different things are mentioned in different languages that aren't mentioned in others."

3.1.5 Cross-Language Plagiarism Detection

Plagiarism detection is a field of research that determines the similarity of two documents with the intention of identifying the theft of language or ideas [Pereira, 2010]. The PAN workshops[7] from 2009 to 2014 have held competitions to encourage development of plagiarism detection techniques on common sets of documents. The workshops have two tasks: source retrieval, where all plagiarised documents need to be retrieved for a given document; and text alignment, where all occurrences of plagiarised text in a pair of documents are identified. The workshops focus on monolingual plagiarism detection.

Cross-language plagiarism detection (CLPD) extends the plagiarism detection research to identifying similar text across different languages [Potthast et al., 2011]. In detecting similar text across languages, translations to a target language are needed, then similarity techniques based on characters, vocabulary, alignments of text, and others are applied to determine similarity scores [Potthast et al., 2011]. The techniques of plagiarism detection and cross-language plagiarism detection focus on retrieval of documents. Gipp [2014] presents a comprehensive look at plagiarism detection techniques (including CLPD) with a particular focus on citation-based plagiarism seen in academic writing. In Chapter 4, we have a similar aim of determining how similar Wikipedia articles are across languages. However, our focus is on the general similarity of the expressed knowledge in articles in multiple languages, which differs from cross-language plagiarism detection as the focus is on identifying portions of plagiarised text.

The similar names of cross-language have different meanings in our cross-language vandalism detection (CLVD) techniques compared with CLPD techniques. In CLPD, the focus is on retrieving plagiarised documents across languages by using text similarity on translated documents. The correct detection of plagiarism is measured by the probable amount of plagiarism in a text document. Translation of text documents is also a complex task as briefly described in Section 3.1.1, which limits language pairings to English and another language because of the limited availability of parallel text corpora. In CLVD, our aim is to develop vandalism detection techniques that are transferable across languages without needing to translate text documents. The focus is on classification models and their reuse for different language domains, which allows complete pairings of source and target languages for detecting vandalism, as shown in Chapters 5 to 7.

[7] http://pan.webis.de/

Overall, the fields of CLPD and CLVD have different meanings in their term of cross-language, but the common goal of investigating research techniques for languages other than English is highly beneficial for transferring techniques and knowledge across to different languages.

3.1.6 Transfer Learning

Transfer learning is a new learning framework that addresses the problem of insufficient data in the domain of interest by relaxing the conditions that training and testing data must be from the same feature space and drawn from the same distribution [Pan and Yang, 2010]. If knowledge transfer of models is successful, then expensive data labelling and data sourcing problems are avoided. Pan and Yang [2010] survey a variety of transfer learning techniques for classification, regression, and clustering problems.

In the traditional machine learning process, a separate learning system is built for each data set, which is seen in past vandalism detection where research is focused on each language individually [West and Lee, 2011]. In the learning process of transfer learning, the domain of the source task and target task are different. This thesis addresses this research gap in vandalism detection by reusing models across languages in Chapters 5 to 7, and specifically across user types in Chapter 6.

Another research that makes use of transfer learning on Wikipedia is by Chin and Street [2012]. This work focuses specifically on learning vandalism from one article and applying the models to another article. Revisions from the Webis Wikipedia vandalism corpus[8] [Potthast and Gerling, 2007] are segmented and placed into similar clusters. The best performing vandalism classification models built on each cluster are then evaluated on clusters from revisions of two selected English Wikipedia articles.

Overall, in this thesis we have investigated the process of transfer learning with successes. Although the features used are common across languages in Chapters 5 and 6, the language domains are different with different distributions and types of vandalism. Furthermore, transfer learning is successful in Chapter 7, where we use part-of-speech tags from different language domains, but the classifier shows successes in classifying vandalism with tags not from the training domain.

3.2 Understanding Vandalism

Vandalism is a burden on Wikipedia, where its occurrence and work in identifying and reverting it are increasing [Kittur et al., 2007]. The time spent on maintenance work (e.g. reverting vandalism) by users are increasing, which leaves less time for writing articles [Kittur et al., 2007]. By its open nature, vandalism or more general malicious edits have occurred on every Wikipedia article [Viegas et al., 2004].

[8] This vandalism data set is the precursor to the data sets provided by the PAN workshops described in Section 3.4.1, but it contains many incorrect labellings of vandalism as identified by Potthast [2010].

To visually appreciate the complexities in the editing activity of each article, Viegas et al. [2004] create history flow visualisations of the edit history of articles, where the contributions of editors over time are illustrated. Some types of vandalism – such as mass deletes and mass additions – can be clearly identified by visual inspection. However, for automated detection of vandalism, analysis of the article content is needed, which has led to the machine learning research in the following sections of this chapter and generally in this thesis. Although many cases of vandalism are repaired almost immediately (on average 2.1 days, and median of 11.3 minutes) [Kittur et al., 2007], the probability that an article will be vandalised is increasing over time [Priedhorsky et al., 2007].

Identifying and repairing vandalised articles is a maintenance task that can be automated using bots, but the problem of vandals remain. Geiger and Ribes [2010] present an analysis of counter-vandalism activities on Wikipedia, from the detection of vandalism to the administrative work in developing and enforcing policies that lead to banning vandals. The analysis show counter-vandalism tools (see Sections 2.4 and 3.3) – especially bots – are becoming predominant in identifying and repairing vandalism, and identifying problematic users and banning them.

Counter-vandalism bots have proved to be valuable in reducing the workload of users and the exposure time of vandalism. Geiger and Halfaker [2013] study a time period in 2011, where one of the most prolific counter-vandalism bots in the English Wikipedia, ClueBot NG (described in Section 2.4), was not operational for four lengthy and distinct periods of time. During these periods, the revert time of vandalism doubled, but eventually all vandalised articles were repaired by users and ClueBot NG as it became operational again.

The prolific nature of counter-vandalism bots have led to some social problems on Wikipedia in retaining new users (human editors). Halfaker et al. [2011] show that reverts made by experienced users or incorrect identification of vandalism by bots reduce the quantity and quality of work of new users and discourage many new users to return. This problem can be addressed by encouraging experienced users to reach out to new users and teach them why their contributions were reverted and about the procedures of contributing to an article and joining the editing community of each article.

This softer stance on protecting articles by experienced users is encouraged by Snuggle, a user interface designed by Halfaker et al. [2014]. Snuggle is a counter-vandalism and socialisation tool for inspecting edits that allows users to mark edits by their quality and provide feedback to the editor. Snuggle is compared to STiki [West et al., 2010b] (see Section 3.3) in a user study of usage and desirability. These tools show the development in the understanding of vandalism, which have resulted in the refinement of tools and ideas.

Overall, vandalism and why users become vandals are becoming better understood by the Wikipedia communities and researchers. The open collaborative editing nature of Wikipedia naturally attracts vandals, but in the harsh response of experienced users in protecting articles and prolific counter-vandalism bots, new users are being driven away and some long-term users are leaving, which may lead to Wiki-

pedia becoming stagnant and developing poor reputations of its communities and practices. The encouragement of socialisation of users are seeing promising changes in attracting new users, and may create fewer vandals or create new counter-vandals. While this change unfolds, we continue to develop automated techniques of detecting vandalism in this thesis as they are essential to helping Wikipedia maintain order and quality in its articles across its many languages.

3.3 Research on Counter-Vandalism Tools

In Section 2.4, we showed the importance of counter-vandalism tools to the Wikipedia community by helping streamline the task of identifying and repairing vandalism. The two types of counter-vandalism tools are bots (automated algorithms) and applications (user interfaces), where few are backed by research. This section is mainly a basis for Chapter 6, where we study the contributions of bots (bot editors) and users (human editors) in the task of repairing vandalism.

Bots are an integral part of Wikipedia because they provide automation to repetitive and mundane tasks, but their contributions are often ignored in research or by the Wikipedia community [Geiger, 2011]. For example, activities of some bots do not appear on the list of recent changes provided by Wikipedia [Geiger, 2011]. The prolific editing activity of bots and their discreetness have led to mistrust by some users because the perceived aggressiveness of bots in completing their task without regards to the social dynamics of the editing communities surrounding each article [Geiger, 2011]. Steiner [2014] has developed an application that shows and compares the editing activity of bots and users in real-time for all Wikipedia languages. The application visualises differences in the editing activity of bots throughout the day, where bots often dominate in editing activity in many Wikipedia language editions. Interruptions of bots in tasks such as detecting vandalism can greatly increase exposure and longevity of vandalism, but they also show the resilience of Wikipedia to eventually restore order [Geiger and Halfaker, 2013]. The importance of bots to Wikipedia is seen through their editing contributions and their influence on the editing culture of Wikipedia through interactions with users across many languages of Wikipedia [Halfaker and Riedl, 2012].

Counter-vandalism bots also suffer a backlash from users (see Section 6.7), which could be attributed to incorrect identification of vandalism. These bots – in particular, ClueBot and ClueBot NG – are evolving to use machine learning techniques to detect more sophisticated forms of vandalism, which takes time to learn correct cases of vandalism. Further mistrust of these bots may stem from their lack of reporting their success rates in detecting vandalism, nor the false positive feedback from users flagging incorrect classifications. To date, we are not aware of any counter-vandalism bots that are backed by published research. Other counter-vandalism bots not described here can be found on Wikipedia[9].

[9] https://en.wikipedia.org/wiki/Category:Wikipedia_anti-vandal_bots

Counter-vandalism applications are changing their design to guide users in identifying and softer handling of potential vandalism cases from incoming edits. One of the most popular editing application is Huggle, which provides an interface for fast browsing of article diffs and reverting of an edit. From studying Huggle and other user interfaces for inspecting edits, two notable applications developed from research are STiki [Adler and de Alfaro, 2007], developed from research on user reputation for vandalism detection; and Snuggle [Halfaker et al., 2014], developed through research on user interface design and socialisation of bots on Wikipedia. Further descriptions of these applications are available in Section 2.4 and other applications not covered can be found on Wikipedia[10]. Overall, these applications are designed for fast inspecting and flagging of revisions within a language, and available mainly in English.

3.4 Vandalism Data Sets for Research

In this section, we survey research using the two types of vandalism data sets detailed in Section 2.5: the manually inspected PAN Wikipedia vandalism data sets from the PAN workshops (Section 2.5.1), and automatically generated Wikipedia vandalism repairs data sets (Section 2.5.2). This section forms a basis for Chapters 4 to 7.

3.4.1 PAN Wikipedia Vandalism Data Sets

The PAN workshops in 2010 and 2011 held competitions to encourage the development of machine learning based vandalism detection techniques. The 2010 competition released the PAN-WVC-10 data set, where Mola-Velasco [2010] won first place by using a set of 21 features to detect vandalism. Adler et al. [2011] later improved on this winning entry by developing additional metadata, text, user reputation, and language features, totalling 37 features. These features are evaluated individually and in combinations using a Random Forest classifier, where using all features show the best performance. Similarly, Javanmardi et al. [2011] further improved the classification results of Mola-Velasco [2010] and Adler et al. [2011] by gathering, developing, and applying a feature reduction technique on a total of 66 features. Javanmardi et al. [2011] also explored combinations of types of features to determine the best feature sets for vandalism detection.

Other techniques showing improvements to the winner of the 2010 PAN workshop focused on analysing other properties of the revision content for vandalism. Wu et al. [2010] presents a text-stability approach to find increasingly sophisticated vandalism. This technique builds on ideas presented in Adler and de Alfaro [2007] on the longevity of words over time to determine the probability that parts of an article will be modified by a normal or vandal edit. Wang and McKeown [2010] use natural language processing to extract syntactic and semantic features, in particular looking at syntax, and n-grams to model topics and semantics to isolate vandalism. Chin

[10] https://en.wikipedia.org/wiki/Category:Wikipedia_counter-vandalism_tools

et al. [2010] statistically model words used in revisions and learn vandalism using an active learning approach. A Wikipedia taxonomy of editor actions is presented, where changes made in the revisions are categorised. Harpalani et al. [2011] improve vandalism detection by analysing stylistic features of text. By characterising authors by linguistic behaviour, vandals may have a unique style of writing. Sumbana et al. [2012] use active learning on samples of vandalised and normal revisions. This technique reduces the need to train on the full PAN-WVC-10 data set. The main drawback of these other techniques is that they are not scalable to the full Wikipedia data sets because of the deep text and structure analysis required to generate their proposed features.

For the PAN-WVC-11 data sets released by the 2011 competition, West and Lee [2011] won first place (for each language: English, German, and Spanish) by developing 65 features for a Random Forest classifier that included many of the features from the entries for the 2010 PAN workshop. These features are described generally as language independent, ex post facto (developed after recognition of vandalism), and language driven features. However, classification in non-English Wikipedia revisions showed very poor performance in the AUC-PR scores (0.708 for German, and 0.489 for Spanish) compared to English Wikipedia revisions (0.822), but comparable performance in the AUC-ROC scores (0.969 for German, 0.868 for Spanish, and 0.953 for English). Cross-language learning of vandalism is not performed, to which this thesis fills the research gap in Chapters 5 to 7.

3.4.2 Wikipedia Vandalism Repairs Data Sets

The construction of data sets in related work is generally similar to our presented data processing method described in Section 2.5.2. These data sets have major differences depending on their purpose, but their construction is now seldom seen since 2010 as the PAN data sets are favoured for the verified vandalism cases. In Section 2.5, we make arguments to use both types of data sets to evaluate detection techniques where possible. Furthermore, most related research papers using this type of data set (including research in Section 3.2) only use and discuss the English editions of Wikipedia.

Smets et al. [2008] use the Simple English Wikipedia and the main English Wikipedia to evaluate vandalism detection techniques based on bag-of-words and Naive Bayes, and Probabilistic Sequence Modeling. The classifiers are evaluated on these data sets and the rule-based counter-vandalism bot ClueBot is also modified to evaluate and compare its performance. Smets et al. [2008] compare differences in the classification scores of the proposed machine learning algorithms and rule-based Cluebot, then present arguments for the need of machine learning for the vandalism detection task.

Features extracted from the metadata of revisions allow all Wikipedia article revisions to be processed because of their relative simplicity compared to the revision content. West et al. [2010a] explore a variety of features generated from the meta-

data of all Wikipedia article revisions for detecting vandalism through rollbacks[11]. The reputation features extracted for articles, users, categories, and countries show interesting variations and sources of vandalism. A data set of rollbacks is extracted from the English Wikipedia, where features are derived and evaluated by an SVM classifier.

Other vandalism detection research that does not fit into other categories in this chapter are based on compression and word analysis. Itakura and Clarke [2009] propose a featureless compression method based on dynamic Markov compression to detect vandalism on Wikipedia. This technique is evaluated on a random selection of articles from the English Wikipedia, where results improves on Potthast et al. [2008] and Smets et al. [2008], suggesting compression algorithms may be able to compete with feature-based techniques. Unfortunately, we cannot find follow up studies nor other compression based vandalism detection methods for comparison. Rzeszotarski and Kittur [2012] propose a longevity of words approach to determine the likelihood a revision will be reverted. The study compared the classification scores of three classifiers (Naive Bayes, SVM, and Random Forest) on a small set of 150 articles from the English Wikipedia. The differences in vandalised revisions, and revisions made by bots and users are also compared and discussed. Rzeszotarski and Kittur [2012] show that words can be a predictor of whether an article revision will be reverted, as newly introduced words are less likely to be related to the article. This study seems to be broadening the work by Adler and de Alfaro [2007] on using the longevity of words as an indicator for vandalism, but not directly considering the context and the dependent nature of words in sentences.

3.5 Context-Aware Vandalism Detection Techniques

There are a few context-aware vandalism detection techniques developed for Wikipedia. We hypothesise that the reason is feature-based techniques have been highly successful in detecting vandalism and they are relatively simpler to develop and tune. We show in Chapter 7 that context-aware detection techniques identify different vandalism cases to feature-based techniques. These techniques have many opportunities for future research as the abundance of features is making new features (effective in distinguishing vandalism) continually more difficult to develop. Furthermore, context-aware techniques can target types of sneaky vandalism, which are difficult types of vandalism that involve changing the meaning of sentences by modifying words.

There are two research papers that address context in detecting vandalism on Wikipedia. Wu et al. [2010] present a text-stability approach to find increasingly sophisticated vandalism. This approach builds on ideas presented in Adler and de Alfaro [2007] on the longevity of words over time to determine the probability that parts of an article will be modified by a normal or vandal edit. Ramaswamy et al.

[11] A rollback is a privileged user permission that allows a relatively small number of users to quickly revert edits made. http://en.wikipedia.org/wiki/Wikipedia:Rollback

[2013] propose two metrics that measure the likelihood of words contributed in an edit of a Wikipedia article belonging to that article with respect to the content and topic. A reduced set of key word pairs – built from the article title and introductory paragraphs, and the changes made by the edit – is sent to the Bing Web search engine to determine the number of Web pages that contains each pair. The proposed measures are not related nor compared to the normalised Google distance [Cilibrasi and Vitanyi, 2007], which is a well-known metric for determining semantic similarity using a Web search engine. Both methods are evaluated using the revision histories (similar data processing to Section 2.5.2) of sampled articles from the PAN-WVC-10 data set because of the numerous words and word pairs resulting from data processing. Our work in Chapter 7 presents a feasible approach to context-aware vandalism detection using part-of-speech (POS) tagging with demonstrative evaluation on the full Wikipedia vandalism repairs data sets and all PAN data sets.

3.6 Summary

In this chapter, we have surveyed published research that had a focus on multilingual aspects and vandalism detection on Wikipedia. The common Wikipedia language across all surveyed research papers is English. Beyond the English Wikipedia, the chosen languages for multilingual study are at the discretion of the researchers. For vandalism detection, the English Wikipedia provides the largest source of vandalism cases and vandalism research (see Section 2.5), which is helped by the manually inspected PAN workshop data sets. Feature-based techniques are the most common for vandalism detection because feature engineering is readily understood, where the main challenge is to develop different types of features that can effectively distinguish vandalism. The emergence of context-aware techniques provide new avenues for research that differ from feature-based techniques and targets more difficult types of vandalism to detect. Another key aspect of multilingual and vandalism research is studying the participation of different types of users (bot, anonymous, or registered) as each have vastly different roles and contributions to Wikipedia. Overall, Wikipedia remains a vast data source for research, where this thesis provides novel contributions that seek to extend vandalism research in the English Wikipedia to other languages.

The next chapter begins our research contributions of this thesis by exploring measures to summarise the vast number of articles on Wikipedia within and across languages. We propose measures to summarise two aspects of Wikipedia articles in multiple languages: their semantic coverage across languages, meaning how similar knowledge is represented in different language editions of the same article; and the activity (or stability) of articles within a language, which allows us to estimate the convergence, stagnation, or saturation of knowledge in an article. These two measures combined summarises a range of article growth patterns over time and may allow us to identify or recommend articles most suited to different human translation skills in future work.

Chapter 4

Coverage and Activity of Wikipedia Articles across Languages

In this chapter, we propose measures of the semantic coverage (similarity) of articles on Wikipedia across languages, and of the activity of articles over time. Wikipedia is a comprehensive online encyclopedia available in many languages, but its growth in number of articles and article content is slowing, especially within the English Wikipedia. This presents opportunities for multilingual Wikipedia editors to apply their translation skills. Assuming ideas are represented similarly across languages, we present activity measures of articles over time, and content similarity measures between multiple languages. With these measures, we can determine article quality and identify Wikipedia articles most suited to different human translation skills. These measures aim to summarise a range of article growth patterns over time.

Parts of this chapter has been published in Tran and Christen [2013b] with extensions for this thesis of additional measures, visualisations, and analyses. We begin by introducing our reasons for summarising Wikipedia data across languages in Section 4.1 and then describe and present statistics in Section 4.2 for the Wikipedia revisions data set for English (en) and German (de). Section 4.3 describes and evaluates the Moses machine translator for use in determining how similar articles are between two languages. Section 4.4 defines the similarity measures and Section 4.5 defines the activity measures. Section 4.6 details our application and evaluation of the proposed measures, and Section 4.7 discusses their significance, quality, and limitations. Finally, we conclude this chapter in Section 4.8 with an outlook to future work.

4.1 Introduction

The growth of Wikipedia since 2007 in number of articles and number of editors has slowed significantly across all languages, especially English [Suh et al., 2009]. The English Wikipedia is currently the largest compared to the other 285+ languages in number of articles. While researchers determine why growth is slowing and ways to

increase growth[1][2][3] [Suh et al., 2009], we see an opportunity for Wikipedia to transform into a more complete multilingual resource.

The importance of language is seen in its diversity and knowledge that shapes our thought and perception [Boroditsky, 2011]. In Wikipedia, this knowledge is captured by communities of 178,831 (1.8% of English and 14.1% of German) registered and 713,427 (1.3% of English and 5.7% of German) anonymous editors working in two languages (Table 4.1). The two largest[4] Wikipedias: English (en) and German (de), have 624,016 (16.7% of English and 50.5% of German) common articles (Table 4.1). The translation efforts on Wikipedia are limited by the changing nature of articles such as content, structure, language, translation tools, and multilingual editors and their interests. To directly help improve translation efforts, Wikipedia research has developed tools to assist editors with the translation task.

This research can assist multilingual editors by providing information on the fluidity of article content, and the semantic coverage of content between languages. To measure these properties, we focus on two highly relevant measures: activity and similarity. Stable articles have low recent activity or low content change. The stableness of an article is difficult to define – and we have not found a suitable candidate in the literature – because of many different interpretations and properties. The similarity of an article between two languages is also difficult to define for the same reasons. We apply well known similarity measures used in information retrieval, text mining, data matching, entity resolution, and topic modelling [Manning et al., 2008; Yeung et al., 2011; Christen, 2012]. With the assistance of the Moses statistical machine translator [Koehn et al., 2007], we have translated all 624,016 bidirectional linked English and German Wikipedia articles to evaluate the proposed measures.

The assumption of this research is that while Wikipedia articles may vary in the way they are written in their respective languages, the knowledge they cover is similar with regards to the words used to convey ideas, concepts, and facts. Based on this assumption, the structure of articles, such as sentence, layout or order, are not relevant means to determine knowledge coverage. While this may not be true for all languages and bilingual speakers, Boroditsky [2011] and Ameel et al. [2009] suggest there may be evidence for this. Ameel et al. [2009] also suggest bilingual lexicons used by bilingual speakers can converge.

Our contributions in this chapter are (1) exploring the use of common measures of similarity for comparing multilingual documents, (2) proposing novel document activity measures, and (3) a large scale application to a multilingual document collection. These measures combined contribute to a ranking of collaborative multilingual documents without analysing every document revision in detail. We discuss the applications possible for combinations of low and high activity and similarity measures as a guide for future work.

[1] http://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia's_growth
[2] http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia
[3] http://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia_extended_growth
[4] As of 1 July 2012 from https://meta.wikimedia.org/wiki/List_of_Wikipedias.

Table 4.1: Basic statistics of data sets

Data set	All articles	Encyclopedic articles	All revisions	Unique usernames	Unique IP addresses
English (en)	12,389,353	3,736,370	305,821,091	10,025,768	55,042,902
German (de)	2,826,811	1,235,009	65,732,032	1,262,688	12,511,832
Common	-	624,016 (en 16.7%) (de 50.5%)	-	178,831 (en 1.8%) (de 14.2%)	713,427 (en 1.3%) (de 5.7%)

4.2 Wikipedia Data Sets

The study in this chapter was completed earlier than the following chapters, and so we present results from the Wikipedia (complete edit history) data dump of 1 June 2012 for the English Wikipedia, and 3 June 2012 for the German Wikipedia. Due the costly nature of processing and the limited availability of translation data sets for other languages, this study is limited to only the English and German languages on Wikipedia. We intend to extend this work in the future as other translation data sets suitable for our translator are made available.

In this chapter, we do not study the effects of vandalism, but the cross-language exchange and stabilisation of knowledge. In particular, we investigate the cross-language aspects of Wikipedia before moving into vandalism detection. This chapter provides an understanding of the cross-language similarities and differences between Wikipedia languages, and provides justification for the need of cross-language research, and the cross-language features and techniques proposed in the later chapters.

Table 4.1 shows some basic statistics of the data set, where common articles are articles in both languages that have returning inter-language links. There is a small number of articles with no returning link from the language they link to. The common registered editors found in both Wikipedias are 0.56% of the English editors, and 6.6% of the German editors.

The count of the usernames are of registered editors who have made an edit on Wikipedia that is recorded in the data dump. There are discrepancies with reported registered users[5] because of blocked users, non-editing registered users, deleted articles and their history, hidden special Wikipedia articles, and other factors. Anonymous editors contribute significantly to Wikipedia, so we include our count of unique IP addresses. Note that the common usernames and IP addresses may not be the same person because the single unified login policy was only recently completed in 2015[6] on Wikipedia and IP addresses are often shared.

For this research, we are interested in the encyclopedic articles, and their semantic coverage across languages. In our data processing step, we removed articles that do

[5] https://en.wikipedia.org/wiki/Special:Statistics
[6] http://www.mediawiki.org/wiki/SUL_finalisation

not conform to the Wikipedia markup[7] and articles with mismatching {·}, {{·}}, [·], and [[·]] tags. This is approximately 1.95% of English, and 1.2% of German articles, of the "Encyclopedic articles" column in Table 4.1.

Other types of articles that are not useful for semantic coverage are redirect articles, and disambiguation articles. We reasoned from manual inspections that disambiguation articles could have biased the results as they usually contain little content and many nouns. The "Encyclopedic articles" column in Table 4.1 shows the final count of articles after removal of articles described above. We tokenise the content of these articles and translated all 624,016 common articles using the Moses translator for analysis.

To simplify descriptions of similarity and activity measures, Table 4.2 describes the important notations used in later sections. Note that $|A_{en}| = |A_{de}| = |A_{en\text{-}de}| = |A_{de\text{-}en}| = |C| = 624,016$.

4.3 Moses Machine Translator

The Moses translator [Koehn et al., 2007] is an offline free and open source statistical machine translator. We use Moses to determine the semantic coverage of two articles from two different languages. We use Moses because we need a translation tool that can handle large numbers of documents without monetary charge like services offered by Google Translate and the Bing Translator. Moses also has an active community of developers with helpful guides, tools, and workshops.

We build a baseline Moses system as instructed by their guides[8], using the (free and open source) IRST LM Toolkit [Federico et al., 2008], and the free ready built Europarl Parallel Corpus [Koehn, 2005] suitable for Moses. We do not attempt to improve the baseline Moses system. We translated the testing data sets (news-test2010 and news-test2011) of approximate 3,000 lines using Google Translate[9] and Bing Translator[10] for comparison. Both are statistical machine translation systems and presumably trained on a much larger and more diverse data set.

The Europarl[11] data set consists of extracted sentences from the European parliamentary proceedings from 1996-2011. We train Moses on the German-English Europarl parallel corpus. Two examples of translations by Moses are given in Tables 4.3 (Example 1) and 4.4 (Example 2). We see that the translation is of high quality for Example 1 (Table 4.3) because of similarities in the simple text and text structure written in both languages. In Example 2 (Table 4.4), the article text in their original languages is written differently by the authors. However, we see in the translations (and the original text) that the general knowledge of that paragraph is represented similarly. These differences show the quality of translation algorithms and training data, and how knowledge is represented differently in each language.

[7] https://en.wikipedia.org/wiki/Help:Wiki_markup

[8] Instructions and build as of 1 June 2012. http://www.statmt.org/moses/?n=Moses.Baseline.

[9] http://translate.google.com/

[10] http://www.bing.com/translator/

[11] We used version 7. http://www.statmt.org/europarl/

Table 4.2: Description of notations used in this chapter.

Notation	Description
en, de	Standard language code for English and German, respectively.
A	Set of all Wikipedia articles and their revisions.
$A_{en}, A_{de} \subset A$	Common English and German, respectively, "Encyclopedic articles" in Table 4.1 that have been translated.
$a_{en} \in A_{en}, \quad a_{de} \in A_{de}$	An English and German article with all its revisions.
$a_{de\text{-}en} = mosesEN(a_{de})$	From Section 4.3, $mosesEN()$ is a function that takes an article in any language and returns the article translated to English.
$a_{en\text{-}de} = mosesDE(a_{en})$	Similarly, $mosesDE()$ returns the article translated to German.
$a_{de\text{-}en} \in A_{de\text{-}en}$	The set of articles translated to English.
$a_{en\text{-}de} \in A_{de\text{-}en}$	The set of articles translated to German.
$C \subset (A_{en}, A_{de\text{-}en}, A_{de}, A_{en\text{-}de})$	Common "Encyclopedic articles" and their translations, accounting for differences in names across languages.
$c_{i,j} \in C$	Article i common to both languages with its translations. Each article i has a revision j. The complete article including all markups is recorded for every revision. There are four versions of $c_{i,j}$, but we ignore the languages here for clarity and introduce when necessary as an additional subscript, e.g. $c_{en,i,j}$.
$0 \leq i < n$	n is the number of articles.
$0 \leq j < m_i$	m_i is the latest revision and also the number of revisions for article i.
$t_{i,j} = time(c_{i,j})$	The time of the revisions in seconds since Unix epoch. Revisions appear in order of time: $t_{i,j} < t_{i,j+1}$.
$s_{i,j} = size(c_{i,j})$	The size of the revisions in bytes.
$\mathbf{t}_i = (\ldots, t_{i,j}, \ldots)$	Vector of times of revisions for article i. Similarly for $\mathbf{s}_i$, vector of sizes.
$\lVert \mathbf{t}_i \rVert$	Euclidean norm of a vector $\mathbf{t}_i$. Similarly for $\mathbf{s}_i$.
$\hat{\mathbf{t}}_i = \frac{\mathbf{t}_i}{\lVert \mathbf{t} \rVert}$	Unit vector of times of revisions. Similarly for $\hat{\mathbf{s}}_i$.
$\mathbf{t}_i \cdot \mathbf{s}_i$	Dot product of vectors $\mathbf{t}_i$ and $\mathbf{s}_i$.

Table 4.3: Example 1: (English title) Treaty of London (1949) | (German title) Londoner Zehnmächtepakt

English	the Treaty of London was signed on May 5 , 1949 , which created the Council of Europe . the original signatories were Belgium , Denmark , France , Republic of Ireland , Italy , Luxembourg , Netherlands , Norway , Sweden and United Kingdom . it is currently referred to as the Statute of the Council of Europe .
German translated to English	the London Zehnmächtepakt , signed on 5 May in 1949 , reasoned the Council of Europe . the signatories were Belgium , Denmark , France , Ireland , Italy , Luxembourg , the Netherlands , Norway , Sweden and Great Britain at the moment . it will be regarded as the Statute of the Council of Europe .
German	der Londoner Zehnmächtepakt , unterschrieben am 5. Mai 1949 , begründete den Europarat . die Unterzeichner waren Belgien , Dänemark , Frankreich , Irland , Italien , Luxemburg , die Niederlande , Norwegen , Schweden und Großbritannien . er wird momentan als die Satzung des Europarats angesehen .
English translated to German	die Vertrag London unterzeichnet wurde im Mai 5 , 1949 geschaffen , die der Rat von Europa . die ursprünglichen Unterzeichner waren Belgien , Dänemark , Frankreich , Republik Irland , Italien , Luxemburg , Niederlande , Norwegen , Schweden und Großbritannien . es ist gegenwärtig als das Statut des Europarates .

To evaluate our translator, we use the BLEU (Bilingual Evaluation Understudy) score [Papineni et al., 2002], a commonly used method of evaluating machine translation system [Callison-Burch et al., 2011]. It measures the similarity of a (machine) translated text to a reference (human translated) text. Highly similar texts have values closer to 1. Note that few humans will achieve a translation score of 1 [Papineni et al., 2002]. Table 4.5 summarises the data set evaluation statistics and BLEU scores.

For comparison, the best BLEU scores are around 0.30 for proposed systems at the Workshops on Statistical Machine Translation[12]. We hypothesise the difference in scores is due to the difference in the language of the training data set and the testing data set, and due to the features in the German language such as compound words, which may contribute to the low scores. Improvements to the translator and data sets are continually seen from the Moses community, where we expect higher scores resulting from future research. For now, the performance of the baseline system is already better in the given testing data sets than the leading online statistical machine translators of Google and Bing, as seen in Table 4.5. With this observation, we see Moses as an alternative to translating large number of articles at no cost, while also providing a reasonable estimate of the semantic coverage of articles.

[12] http://www.statmt.org/

Table 4.4: Example 2: (English title) United Nations Security Council Resolution 3 | (German title) Resolution 3 des UN-Sicherheitsrates

English	United Nations Security Council Resolution 3 , adopted on April 4 , 1946 , acknowledged that the Soviet troops in Iran could not be removed in time to meet their deadline under the tri-partite Treaty but requested the Soviet Union remove them as fast as possible and that no member state in any way retard this process . if any developments threaten the withdrawal of troops , the Security Council requested to be informed .
German translated to English	3 of the UN Security Council resolution was adopted on 4 April 1946 . the Security Council is in the resolution notes that the Soviet troops from iranischem area have not been withdrawn . the Soviet Union will have been asked to withdraw troops as quickly as possible ; all other states , not to obstruct the withdrawal of troops in the event that the withdrawal should be at risk of the Security Council wants to be informed .
German	Resolution 3 des UN-Sicherheitsrats wurde am 4. April 1946 angenommen . der Sicherheitsrat stellt in der Resolution fest , dass die sowjetischen Truppen nicht fristgemäß aus iranischem Gebiet abgezogen wurden . die Sowjetunion wird ersucht , die Truppen so schnell wie möglich abzuziehen ; alle anderen Staaten , den Truppenabzug nicht zu behindern . für den Fall , dass der Truppenabzug gefährdet werden sollte , wünscht der Sicherheitsrat informiert zu werden .
English translated to German	Resolution des Sicherheitsrats der Vereinten Nationen 3 vom April 4 , 1946 anerkannt , dass die sowjetische Truppen im Iran nicht rechtzeitig beseitigt werden , ihre Frist unter der tri-partite Vertrag , sondern um die Sowjetunion so schnell wie möglich beseitigt werden und dass kein Mitgliedstaat in irgendeiner Weise durch dieses Prozesses . wenn irgendwelche Entwicklungen bedrohen den Rückzug der Truppen , der Sicherheitsrat aufgefordert , informiert werden .

Table 4.5: Summary of data sets used by Moses. Higher BLEU scores are better.

Corpus	Sentences	German words	English words	BLEU (de-en)	BLEU (en-de)
German-English (train)	1,920,209	44,548,491	47,818,827	-	-
news-test2008 (tune)	2,051	294,135	262,683	-	-
news-test2010 (Moses)	2,489	377,971	328,251	**0.183**	**0.135**
news-test2010 (Google)	"	"	"	0.164	0.102
news-test2010 (Bing)	"	"	"	0.164	0.095
news-test2011 (Moses)	3,003	443,973	396,574	**0.172**	**0.100**
news-test2011 (Google)	"	"	"	0.134	0.088
news-test2011 (Bing)	"	"	"	0.149	0.087

4.4 Multilingual Similarity

The multilingual similarity of two articles in two different languages is determined by analysing the words in the articles and their translated equivalents. For article pairs $(a_{en}, a_{de\text{-}en}, a_{de}, a_{en\text{-}de}) \in C$, similarity is determined by comparing the pairs a_{en} and $a_{de\text{-}en}$, and a_{de} and $a_{en\text{-}de}$, respectively.

We compare the original to its translated version to determine how complete the semantic coverage of articles is in each language. This may seem odd as articles are likely written by different users in different languages, and with different cultural context. However, for an article to exist in two languages suggests there are ideas, knowledge, and facts that transcends both cultures [Boroditsky, 2011]. Some articles may have different points of view or interpretation in each language, such as historical articles. However, we assume there is a consistency in the terminologies used between languages because of Wikipedia's neutral point of view policy, which limits diversity of opinions and terminologies.

We apply common similarity measures from information retrieval, and entity resolution research. We use the Jaccard index, Dice's coefficient, and the Cosine similarity [Christen, 2012], which have been used to determine similarity for sets of words or entire articles in monolingual and bilingual research [Gabrilovich and Markovitch, 2007; Thomas and Sheth, 2007; Medelyan et al., 2009; Yuncong and Fung, 2010; Mohammadi and Ghasem-Aghaee, 2010; Yeung et al., 2011]. We extend the Cosine similarity with TF-IDF (term frequency-inverse document frequency) [Manning et al., 2008], and show that these methods are more insightful. We cannot use the BLEU score as a similarity measure because the articles in English and German are not sentence aligned. Extracting aligned sentences from Wikipedia is another complex research area beyond the scope of this work [Madnani and Dorr, 2010].

For each current revision in all four versions, $c_{i,m_i} \in C$, we remove stopwords[13] and punctuation relevant to each language, and clean other irrelevant tokens created by Moses. Let this cleaned set of current revisions be $d_i \in D$, where $d_i = clean(c_{i,m_i} \in C)$ and $clean()$ is the function for the described cleaning task.

We use a TF-IDF representation of articles to account for the range of article types and the frequency of occurring words within these articles. The Gensim [Rehurek and Sojka, 2010] topic modelling toolkit is used to calculate the TF-IDF of words for each set A_{en}, A_{de}, $A_{en\text{-}de}$, and $A_{de\text{-}en}$. We remove 10% of the most common words (lowest TF-IDF scores) to remove further potential common words.

We look at three common similarity methods based on sets: Jaccard index, Dice's coefficient, and Cosine similarity [Christen, 2012]. We also introduce a couple of methods using TF-IDF in the Cosine similarity: CosineTF-IDF [Manning et al., 2008], and CosineTF-IDFCombined, which is a modification of CosineTF-IDF. We define the measures below for the target language of English. The same measures are also calculated for the German articles and their translated version from English. For the measures below, $|X|$ is the cardinality of set X.

[13] http://tlt.its.psu.edu/suggestions/international/bylanguage/index.html

Let $wordSet()$ return a set of words in an article. For $d_{en,i}, d_{de\text{-}en,i} \in D$, let $X_i = wordSet(d_{en,i})$ and $Y_i = wordSet(d_{de\text{-}en,i})$. Then, we define the following similarities:

$$sim_{Jaccard,en,i} = \frac{|X_i \cap Y_i|}{|X_i \cup Y_i|} \qquad (4.1) \qquad sim_{Dice,en,i} = \frac{2|X_i \cap Y_i|}{|X_i| + |Y_i|} \qquad (4.2)$$

$$sim_{Cosine,en,i} = \frac{X_i \cdot Y_i}{\|X_i\| \|Y_i\|} \qquad (4.3)$$

Let $wordList()$ return a list of all words in an article. Let $R_i = tf\text{-}idf(wordList(d_{en,i}))$, $S_i = tf\text{-}idf(wordList(d_{de\text{-}en,i}))$, and $T_i = R_i + S_i$, which is the result of appending the two lists. Then, we define the following Cosine similarities with TF-IDF:

$$sim_{CosineTF\text{-}IDF,en,i} = \frac{R_i \cdot S_i}{\|R_i\| \|S_i\|} \qquad (4.4)$$

$$sim_{CosineTF\text{-}IDFCombined,en,i} = \frac{1}{max(T)} \sum_j t_{i,j} \quad , \quad t_{i,j} \in T_i \qquad (4.5)$$

The set measures (Equations 4.1, 4.2, and 4.3) do not account for frequency or importance of words in an article. By taking words in an article as a list, we weight the importance of words by its frequency of occurrence. The measure $sim_{CosineTF\text{-}IDF,en,i}$ (Equation 4.4) aims to further emphasise important words between two languages as they are used in articles. For $sim_{CosineTF\text{-}IDFCombined,en,i}$ (Equation 4.5), the TF-IDF is calculated on the combined corpora of the same language to counteract poor machine translations. This shows the importance of the combined English and German articles relative to their corpora. With this corpora, words that could not be translated become much less important.

4.5 Article Activity

The activity of an article aims to estimate when knowledge in an article has stabilised, converged, become stagnant, or become saturated with all relevant information. We measure the activity (or stability) from the time and size of a revision. We consider stability and activity to be complementary terms. We define measures of activity to avoid confusion of semantics with stability, so highly stable articles have low activity and vice versa. We look at revisions of Wikipedia articles over their entire lifetime and in the current year (2012) to determine activity. We look at the current year to gauge recent activity of articles.

Article activity measures are subjective and difficult to define, so we build on the idea of semantic convergence. We look at revisions of Wikipedia articles over their entire lifetime and in the current year to determine activity. We look at the current year to gauge recent activity of articles. We also introduce a variant to all measures that incorporates geometric decay, which emphasises the importance of more recent revisions. We compare both variants to see what insights they offer.

Thomas and Sheth [2007] applies semantic convergence to Wikipedia. However,

the authors use a small set of thousands of monolingual Wikipedia articles with the aim of recommendation and topic modelling. Thomas and Sheth [2007] defines semantic convergence as when the Cosine distance between TF-IDF represented revisions falls below a threshold. Semantic convergence for the monolingual case has benefits, such as determining and predicting mature articles.

We account for vandalism by removing edits labelled as vandalism by editors (see Section 2.5.2), and others with the comments of mass deletes or mass inserts. On visual inspection of over 100 articles, we find this identifies the majority of obvious vandalism and accounts for reverts made immediately after edits, presumably poor or rejected edits.

4.5.1 Gradient

The semantic convergence of articles suggests there is a saturation point for knowledge in an article. This can be modelled by logarithmic growth, where high activity and high increase in the articles size are seen early in the life of an article. Then activity gradually decreases over a long time period, and the size of the article increases very slowly. This assumption is not true for all articles, but it is a simple measure. To approximate this logarithmic change, we take the natural log of the times and sizes of an article's revisions, then perform a simple linear regression. We use the gradient as our measure of activity, where a lower gradient means fewer semantic changes in the article. Equation 4.6 shows our activity measure based on the gradient of the size of an article with the time of the change. Equation 4.7 adds a geometric decay of $\frac{1}{2^{m_i - j}}$ to emphasise the relatively recent revisions by the time when edits were made.

$$act_{Gradient,en,i} = \frac{2}{\pi} \arctan \left| \frac{n \sum_j \log(t_{i,j}) \log(s_{i,j}) - \sum_j \log(t_{i,j}) \sum_j \log(s_{i,j})}{n \sum_j \left(\log(t_{i,j})\right)^2 - \left(\sum_j \log(t_{i,j})\right)^2} \right| \tag{4.6}$$

$$act_{Gradient(decay),en,i} = \frac{2}{\pi} \arctan \left| \frac{n \sum_j \log(t_{i,j}) \log(s_{i,j}) - \sum_j \log(t_{i,j}) \sum_j \log(s_{i,j})}{n \sum_j \left(\frac{1}{2^{m_i - j}} \log(t_{i,j})\right)^2 - \left(\sum_j \frac{1}{2^{m_i - j}} \log(t_{i,j})\right)^2} \right| \tag{4.7}$$

4.5.2 Relative Change

These vector space measures look at the relative change in size of content of a revision compared with its previous revision. This differentiates the significance of the number of bytes changed compared to the whole article. For example, a change of 100 bytes in an article of 1,000 bytes is much more significant than in an article of 10,000 bytes. Similarly, 1 day between edits is more significant than 10 days as it may indicate high short term activity.

We capture this relative change by calculating for all revisions j of article i, the dot product of unit vectors of the time and size relative difference calculations. Equa-

tion 4.8 shows our proposed measure, where the dot product is also known as the cosine similarity between two vectors. We the cosine similarity to combine and reduce two different measures of activity to a single value. An article is considered to be stabilising (have low activity) when the time between revisions increases and the size between revisions changes little. This means the vectors become more similar over time. By considering only the 2012 revisions or applying geometric decay in Equation 4.9, we select the most recent revisions to gauge the latest activity. These selections further emphasise stable (low activity) articles as they are likely to have long periods of inactivity and little change in content.

$$\mathbf{p}_i = \left(\dots, \left| \frac{t_{i,j+1}}{t_{i,j+1} - t_{i,0}} \right|, \dots \right) \qquad \mathbf{q}_i = \left(\dots, \left| \frac{s_{i,j+1} - s_{i,j}}{s_{i,j+1}} \right|, \dots \right)$$

$$act_{Relative,en,i} = \frac{1}{n} \hat{\mathbf{p}}_i \cdot \hat{\mathbf{q}}_i \tag{4.8}$$

$$\mathbf{r}_i = \left(\dots, \left| \frac{t_{i,j+1}}{2^{m_i - j}(t_{i,j+1} - t_{i,0})} \right|, \dots \right) \qquad \mathbf{s}_i = \left(\dots, \left| \frac{s_{i,j+1} - s_{i,j}}{2^{m_i - j} s_{i,j+1}} \right|, \dots \right)$$

$$act_{Relative(decay),en,i} = \frac{1}{n} \hat{\mathbf{r}}_i \cdot \hat{\mathbf{s}}_i \tag{4.9}$$

4.5.3 Information Entropy

This vector space activity measure looks at the change in the information content between each revision compared to the final revision. We use information theory [Cover and Thomas, 2012] to model entropy as a measure for activity. Articles with low entropy for its revisions means there is low uncertainty with its data, which is interpreted as an estimation of activity in an article.

We apply the entropy measure differently for the endpoints of time and size of revisions in an article because of the unbounded nature of revisions to an article. For time, we choose a fix period of time that spans all articles: t_{start} as the day of the oldest revision in the current Wikipedia database in epoch time, and t_{end} as the day after the Wikipedia data dump completed (for each language) in epoch time; where we have $\Delta t = t_{end} - t_{start}$.

The change in size of an article is relative for each article as there is no "largest" article that can be compared ubiquitously across all articles. For each article, we use s_{i,m_i} as the size of the latest or current revision of the article in the data dump. We assume all articles begin with size 0 for a consistent starting point. This is separate from revision $s_{i,0}$, which has an initial size. Thus, $\Delta s_i = s_{i,m_i}$. Then we have our activity measure based on entropy. For time, we use a fixed period of time as a comparison for change between revisions, which are independent on the articles.

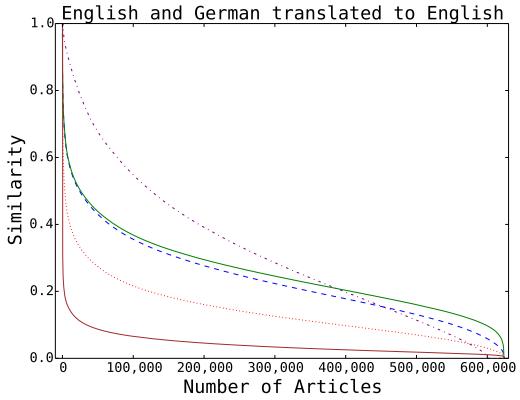

Figure 4.1: Similarity distributions for comparing articles in English, sorted independently of article. 1 is high similarity.

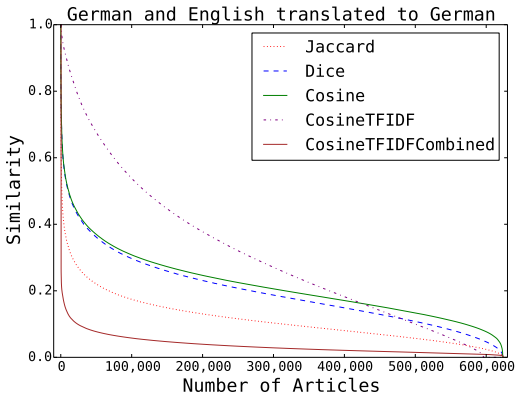

Figure 4.2: Similarity distributions for comparing articles in German, sorted independently of article. 1 is high similarity.

For size, we use a relative change in size to measure change in information content between revisions.

Equation 4.10 defines our proposed Entropy measure, and Equation 4.11 adds a geometric decay to favour recent changes to an article. Similar to the Relative measure, we combine the time and size values by using the cosine similarity.

$$\mathbf{p}_i = \left(\ldots, -log \left| \frac{\Delta t}{t_{end} - t_{i,j}} \right|, \ldots \right) \qquad \mathbf{q}_i = \left(\ldots, -log \left| \frac{s_{i,m_i} - s_{i,j}}{s_{i,m_i}} \right|, \ldots \right)$$

$$act_{Entropy,en,i} = \frac{1}{n} \hat{\mathbf{p}}_i \cdot \hat{\mathbf{q}}_i \tag{4.10}$$

$$\mathbf{r}_i = \left(\ldots, -log \left| \frac{\Delta t}{2^{m_i - j}(t_{end} - t_{i,j})} \right|, \ldots \right) \qquad \mathbf{s}_i = \left(\ldots, -log \left| \frac{s_{i,m_i} - s_{i,j}}{2^{m_i - j} s_{i,m_i}} \right|, \ldots \right)$$

$$act_{Entropy(decay),en,i} = \frac{1}{n} \hat{\mathbf{r}}_i \cdot \hat{\mathbf{s}}_i \tag{4.11}$$

4.6 Evaluation

In this section, we apply the proposed similarity measures to the 624,016 original and translated Wikipedia articles from English and German, present summary statistics, and give descriptions of plots. For similarity, values close to 1 are highly similar articles. For activity, values close to 1 are high activity articles and values close to 0 are highly stable articles.

4.6.1 Cross-Language Similarity

Figures 4.1 and 4.2 show the distribution of similarity values for pairs of articles for the target languages of English and German, respectively. Note that each measure is sorted independently of the article for visual clarity. The three similarity measures based on sets (Jaccard, Dice, and Cosine; Equations 4.1, 4.2, and 4.3, respectively) do show the same ordering of articles. The two proposed similarity measures based on TF-IDF (Equations 4.4 and 4.5) showed lower similarity values. There are differences between the two graphs, but because of the similar shapes of the distributions, the differences may be difficult to see. We considered adding gridlines for clarity, but this cluttered the plots significantly.

The similarity distribution of the set based measures show very similar shapes in addition to the same ordering of documents. This suggests information has been lost because of the assumed equal distribution of words in each article and its translated counterpart. The lower overall similarity distribution in the German measures may be caused by features of each language not captured by the translator. In particular, English words may not be mapping correctly to German compound words. This may be most apparent in the least similar articles. The low sloping of the set measures suggests most articles have the same basic set of common terminologies.

Representing words in an article as a TF-IDF vector produces very different results to assumed equal distribution. For both target languages, the CosineTF-IDF measure shows a different shape compared to the other set measures. The similarity distribution is spread more evenly across articles. This allows a better distinction of similar articles across languages. Note the effect of incorrect mapping of English words to German compound words may have a less significant effect because TF-IDF automatically accounts for those words (as they are likely to be rare occurrences) when comparing to each corpus of A_{en} and A_{de}.

The second TF-IDF measure, CosineTF-IDFCombined, does not provide good discernible information for the articles because of many skewed low values. By combining the original article and its translated counterparts into the same domain, the effect of highly important words dominate other words. This measure provides a good similarity measure for articles with many important words with respect to the combined corpora.

4.6.2 Article Activity

The activity of articles is calculated from revisions of the common articles. Figures 4.3 to 4.6 show the activity measures applied to each article, sorted independently of article for visual clarity. All the measures show different distribution types, where the decay measures show a slight difference to the non-decay measures. With the Gradient measures, the addition of decay has a strong effect, where the decay version ceases to show useful information, indicating very few recent activity. The Relative measures do not show much difference with or without decay. The Entropy measures show a gradual decline in activity for 2012 revisions, but a much steeper decline over

the lifetime of articles compared to the Relative measures. The Entropy measure with decay show lower values of activity than the non-decay measure.

For activity, we are interested in articles with low activity. With the 2012 revisions, more than a third of English and half of German articles are considered to be highly stable by all measures. Considering all revisions of all translated articles, the majority of articles have high activity with respect to all measures except Gradient. A reason for the Gradient measure overestimating is the short lifespan of most articles, which gives a steeper slope in $act_{Gradient,i}$ (Equation 4.6).

The distributions have their distinctive shapes, where the Entropy measures are distinct from the other two types of measures. However, the shape of the distributions remains similar across languages, suggesting similar types of editing activity. The Gradient and Relative Change measures have a distinctive plateau, suggesting many articles are experiencing few editing activity. A small proportion of articles are highly unstable, with many edits over its lifetime and in the 2012 revisions.

From Figures 4.3 and 4.4, the Entropy measures for the 2012 revisions suggest similar types of editing activity, with a more gradual change in the information content of articles. Figures 4.5 and 4.6 show different shapes for the Entropy measures over all revisions. These suggest over the lifetime of an article, small changes frequently occur over a large set of articles, and the English Wikipedia seems to have a higher overall level of activity. The majority of articles are stable or highly stable (low level of activity) for the 2012 time period, presenting opportunities for multilingual editors to apply their skills.

4.6.3 Similarity and Activity

We have summarised the semantic coverage of (current revisions of) articles with similarity measures and semantic convergence of articles (from all revisions) with activity measures. Combining these two types of measures shows a greater range of article types and information. Figures 4.7 to 4.9 show the combination of the CosineTF-IDF measure with all the presented activity measures for the 2012 revisions. We show only 2012 revisions because the most recent revisions may be a better indicator of the activity of articles. We choose the CosineTF-IDF measure because of its quality and meaning compared with the other measures. The other similarity measures show different distributions of values across different activity measures. Although we do not include them for brevity, CosineTF-IDF represents the similarity measures well. The patterns of the activity measures are more prominent with these plots.

The Gradient measures in Figure 4.7 show dense regions of articles with high activity but low similarity across English and German. In the Gradient decay measure, the preference for the most recent revisions show very few articles with activity because the gradual change in Equation 4.6 is filtered strongly in the decay equation of Equation 4.7. Thus, most values of activity is zero or very close to zero for any meaningful interpretation.

The Relative measures in Figure 4.8 show a range of articles towards the low

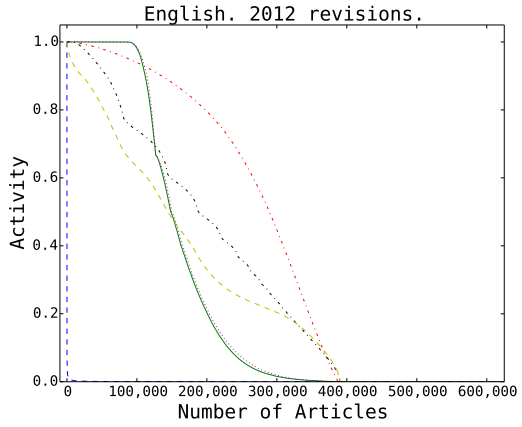

Figure 4.3: Activity distributions for 2012 revisions for English, plotted independently of article for clarity. 1 is high activity.

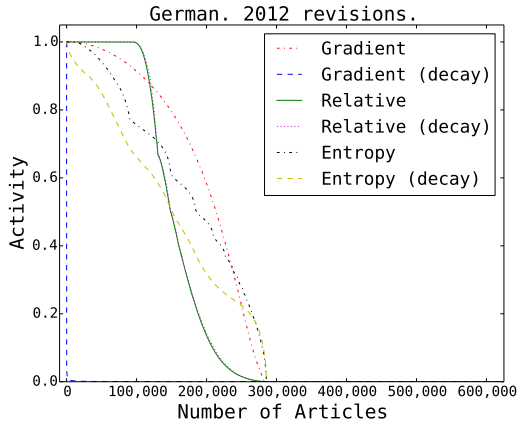

Figure 4.4: Activity distributions for 2012 revisions for German, plotted independently of article for clarity. 1 is high activity.

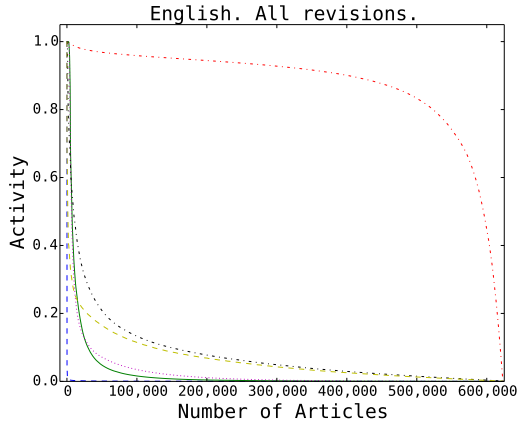

Figure 4.5: Activity distributions for all revisions for English, plotted independently of article for clarity. 1 is high activity.

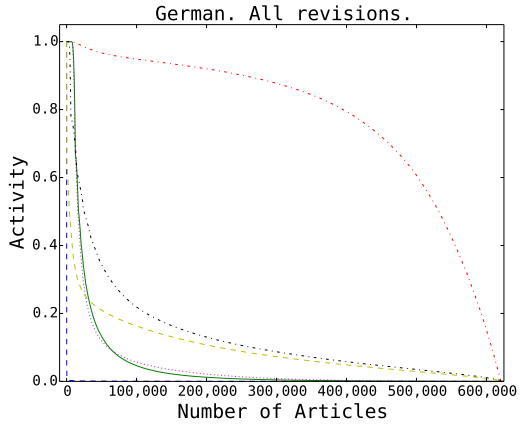

Figure 4.6: Activity distributions for all revisions for German, plotted independently of article for clarity. 1 is high activity.

similarity side, all with distinct levels of activity. The Relative measure does not show many visual differences with or without decay. This suggests recent activity on the articles make up most of the activity in the 2012 revisions, which explains the lack of differences. The distinct bands of activity and difference in the maximum result (activity axis) are the result of the cosine similarity of the normalised vectors in Equations 4.8 (no decay) and 4.9 (with decay). These bands suggest cycles of activity patterns from the relationship between changes in the size of articles and when changes are made.

The Entropy measures in Figure 4.9 also show the bands of activity similar to the Relative measures but with different cycles. From Equation 4.10, the logarithmic

values further discriminates low levels of activity compared to the Relative measures. We may be observing articles with low level cycles of activity in the non-decay heatmap. With the addition of decay, we see different bands of very recent activity in articles. The lowest activity band showing a dense number of articles may be the activity of minor revisions in well knowledge saturated articles. The higher bands may be showing articles requiring different levels of work.

Overall, these plots convey various types of information. In future work, we intend to group articles according to a higher topics (such as sciences, mathematics, arts, and others on Wikipedia[14]) to visualise the differences between articles of different topics. To support multilingual Wikipedians, articles with low activity and low similarity are better candidates to improve similarity, as translations are likely to remain semantically similar between languages for longer periods of time.

4.7 Discussion

The significance of these measures is seen in many applications. For example, similarity is used in ranking, search, information extraction, topic modelling, and linguistics [Manning et al., 2008]; and activity is used to determine quality, validity, completeness, correctness, and maturity of articles [Thomas and Sheth, 2007]. Wikipedia's growth is slowing across all languages, but this presents an opportunity to build a more complete multilingual Wikipedia. These measures provide rankings and identification of different types of articles to suit different translation skills.

For similarity, the extension of common set-based approaches to similarity gives insight into semantic coverage of articles across languages. The evaluation shows a visual difference in the distribution of the TF-IDF based measures, which give a greater spread because the measures account for peculiarities of the words in the corpus. Refocusing these similarity measures on the monolingual history of an article allows evaluation of semantic convergence, which is seen in the literature for Wikipedia. Limitations of past research on semantic coverage across languages are using small evaluation sets of sampled articles [Thomas and Sheth, 2007], particular sentences from sampled articles [Filatova, 2009], or structured properties of the Wikipedia markup [Yeung et al., 2011]. Our similarity measures aim to be scalable, novel in their use of TF-IDF, and informative of semantic coverage of whole article contents.

For the activity measures, these are novel for ever changing documents. While activity is usually subjectively interpreted, we present objective measurements based on three different ideas: the rate of change estimated by simple linear regression, relative changes between revisions, and most interestingly change in entropy over time. From the evaluation of these measures, the entropy measure provides a very different distribution to the other measures. Entropy seems to better capture the activity in articles based on basic numbers of size and change. We find a large proportion of articles with no activity, and a small proportion with very high activity

[14] https://en.wikipedia.org/wiki/Portal:Contents/Portals

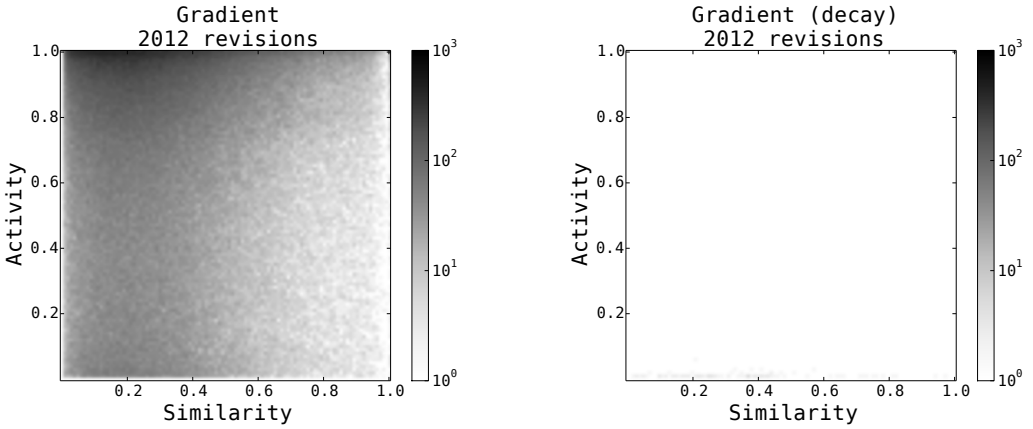

Figure 4.7: Activity (CosineTF-IDF) score of 2012 revisions plotted against similarity (Gradient) scores of all 624,016 articles. The Gradient decay measure has many zero values because of the strong penalty of geometric decay in Equation 4.7.

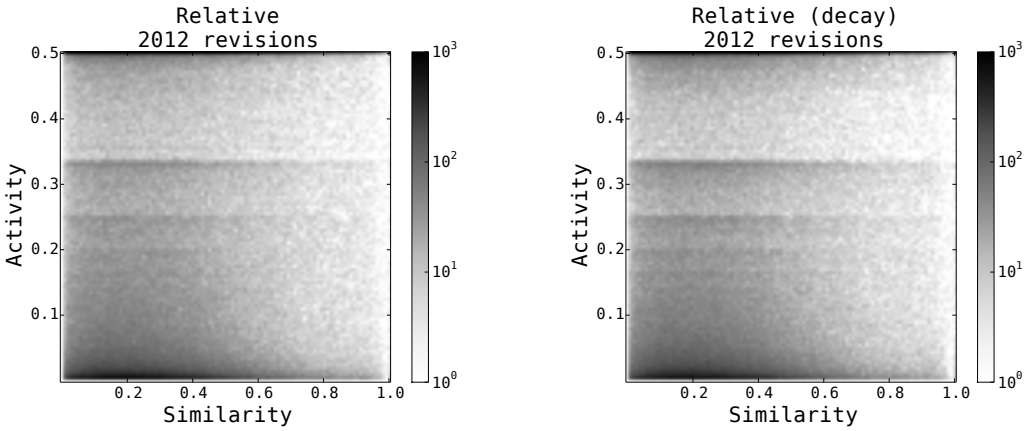

Figure 4.8: Activity (CosineTF-IDF) score of 2012 revisions plotted against similarity (Relative) scores of all 624,016 articles. The distinct bands suggest cycles of activity seen in the relative changes in time and size of articles.

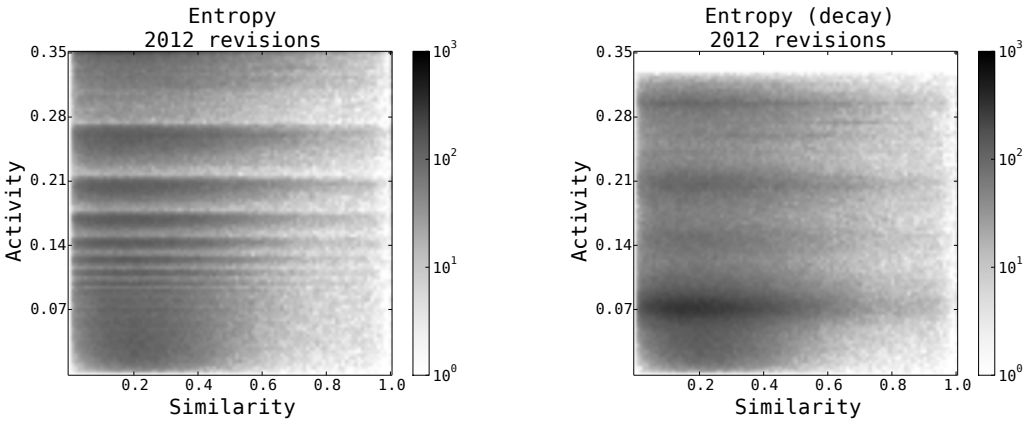

Figure 4.9: Activity (CosineTF-IDF) score of 2012 revisions plotted against similarity (Entropy) scores of all 624,016 articles. The distinct bands suggest cycles of activity similar to Figure 4.8 above, but the activity measures emphasise small changes to articles.

for the 2012 revisions of the data dump. This shows the Wikipedia community is active, but they are focused on a small set of articles.

The quality of articles can be objectively inferred from its similarity and activity measurements. Articles with high similarity across languages suggest a convergence of knowledge representation, which adds to the quality of both articles. With the addition of other languages, high similarity across multiple languages shows an agreement on the knowledge in the article. For activity, the time period is important as long-lived articles with many revisions suggests activity. As an article is refined over time, knowledge in an article approaches a saturation point, where modifications are made, but overall content does not change significantly. Thus, activity shows an agreement on knowledge within the same language domain.

The quality of the measures is difficult to assess as quality is usually a subjective measure. A user study is needed to rate these articles on similarity, which could provide a measure of quality. As in the BLEU score experiments, we compare these measurements to human interpretations. Some measures are very different in ideas, such as the TF-IDF and entropy measures. The simpler measures show similar characteristics and ordering, suggesting similar quality. However, having many similar measures does not provide further information on the articles as seen from the shape of the similarity-activity distributions. Thus, while the combination of TF-IDF and entropy measures can be vastly different from the simpler measures, they offer a different summary view of the similarity and activity of articles.

The key issue for the similarity measures is the use of the Moses translator. The quality of translation is heavily dependent on the set of training data. The availability of the ready to use Europarl data set makes Moses practical, but the language used by the Euro parliamentary proceedings is limited. The language used in Wikipedia is much more general and diverse. Thus, more general training data sets for Moses, and improvements on the baseline algorithm are needed. However, despite these shortcomings, Moses remains an attractive choice because of its open source licensing and active community of developers.

Some types of Wikipedia articles do not provide good measures, such as redirect or disambiguation articles. The complexity of Wikipedia markups also creates hidden complexities, such as disambiguation articles, lists, and stubs. We removed disambiguation articles because they are numerous. There are comparatively fewer lists, where on manual inspection we often find lists containing useful content. We found stub articles are sometimes inconsistently labelled, so we keep them in our analysis. Furthermore, Wikipedia markups between (and within) languages are inconsistent with many hidden features.

German and English are closely related languages. Thus, the semantic coverage in the similarity measure may be optimistic. Further applications to other human language domains are needed. Despite the use of related languages, English and German have enough peculiarities to hinder the Moses translator.

For the activity measures, an extension to consider the content of articles is planned. Determining the change in entropy based on size of revisions and date of revisions seems incomplete. Crucial information such as that provided by TF-IDF

could provide a better measure of entropy.

The Moses translator is the main computational bottleneck as translation of many documents takes a significant proportion of time. Other translation methods such as direct word translation is faster, but they sacrifice semantics in sentences [Hutchins and Somers, 1992]. Sentence structure plays a crucial role in machine translation to determine types of meaning. However, a simple translation of words may suffice for the TF-IDF measures. In this research, we seek a complete translation of all common articles for completeness, but importantly the data set contributes to collaborative translation systems, user studies, and an interesting data set for research.

We made the claim of scalability because of our application to over 600,000 articles. We initially had progressive scaling in increments of 100,000 articles, but we deemed a full summary was easier to interpret. We believe the application to 600,000 articles is sufficient evidence. Although we do not have the timings of application because of the parallel nature of our statistical code, our design ensure that all our measures are O(n) time, where n is the number of articles.

4.8 Summary

In this chapter, we have presented sets of similarity and novel activity measures for ever changing multilingual articles. We applied these measures to 624,016 common articles from the two Wikipedia language editions: English and German. We showed the effectiveness of these measures, and compared them to common or novel proposed measures. These measures are scalable, objective, and informative for Wikipedia editors and various types of research. Our main assumption is that ideas, knowledge, and facts are represented similarly across languages because of language effects and particularly Wikipedia's neutral point of view policy, which limits diversity of opinion. We used Moses, a machine translator, to help with the similarity task to determine semantic coverage between two languages. These measures can be readily used in other research and applications, but foremost to aid the rare multilingual Wikipedia editors, and to lower barriers for new multilingual editors by objectively providing articles suited to their translation skills.

In future work, we intend to use these similarity and activity measures as additional features in tasks such as topic modelling, classification, and prediction and identification of articles. These tasks look to exploit these measures to discern the types of articles that are stable and how they become stable, and how similarity is achieved across human language domains. Further improvements are possible to the Moses translator with more comprehensive training data. English and German are closely related languages, so there are possible cross-language effects beyond English by comparing different families of languages as high quality translation data sets become available. Our novel use of TF-IDF and information entropy need further exploration, possibly beyond Wikipedia into other types of ever changing documents such as source code. We plan to extend these similarity and activity measures beyond Wikipedia to demonstrate their practicality and effectiveness in other domains.

The next chapter begins our development of cross-language vandalism detection (CLVD) techniques. We start with developing features from the metadata of two data sets: the revision history data set that is commonly used in research, and an article views data set never before used for vandalism detection. We show how cross-language learning is performed and the benefits it has for non-English and small Wikipedias, where there are comparatively few research papers compared to research on the English Wikipedia.

Chapter 5

Metadata Features for Cross-Language Vandalism Detection

In this chapter, we evaluate a range of classifiers to assess their feasibility for cross-language vandalism detection. We focus on the metadata features of two data sets for two languages (English and German) to gauge the effectiveness of cross-language learning. The Wikipedia article views data set is rarely used for research and has never before been used for vandalism detection. We demonstrate how enriching the article views data set with features from the article revisions data set can improve detection of vandalism. This suggests the viewing statistics of vandalised articles may be changing due to factors such as vandals revisiting frequently to check their work or increases in interest as people share vandalism online or with their friends. The key advantage of using metadata features is language independent features are simple to extract as they require minimal processing. This chapter shows cross-language application of vandalism models is possible, and vandalism can be detected through view patterns of articles.

This chapter has been published in Tran and Christen [2013a] and is organised as follows. Section 5.1 introduces vandalism detection from metadata and the need for cross-language classification. Section 5.2 describes the Wikipedia article revisions data set and the article views data set, and how to create the combined data set. Section 5.3 details the machine learning algorithms to be compared for cross-language vandalism detection. Section 5.4 summarises the results by providing AUC-PR and AUC-ROC scores, and approximate execution times. Section 5.5 discusses the significance, quality, and limitations of this data set and approach. Finally, we conclude this chapter in Section 5.6 with outlook to future work.

5.1 Introduction

Vandalism is a major issue on Wikipedia, where the majority of vandalism is caused by humans that can leave traces of their malicious behaviour through access and edit logs. We hypothesise that vandalism can be characterised by the view patterns of

vandalised articles. Vandals may be eliciting behavioural patterns before, during, and after a vandalised edit. We acknowledge similar work by West [2013], where features were developed to capture an article's popularity for the purpose of detecting spam on Wikipedia. We further hypothesise that behaviour of vandals is similar across language domains. This means models developed in one language can be applied to other languages. This can potentially reduce the cost of training classifiers for each language. We find this cross-language application of vandalism models produces similarly high results as for a single language.

In this chapter, we explore cross-language vandalism detection by using a relatively unexplored data set, the hourly article view count, and the commonly used complete edit history of Wikipedia (as described in Section 2.5). We also combine these two data sets to observe any benefits from additional language independent features. We look at two language editions, English and German, and compare and contrast the performance of standard classifiers in identifying vandalism within a language and applied across languages.

Our contributions are (1) novel use of the hourly article views data set for vandalism detection; (2) creation and combination of data sets with language independent features; and (3) showing the cross-language applicability of vandalism models built for one language.

5.2 Wikipedia Data Sets

We use two data sets: the complete edit history of Wikipedia in English and German[1], and the hourly article view count[2]. We describe data with language codes "en" for English and "de" for German. These two raw data sets are processed as described in the subsections below. Due to the overwhelming size, download and processing time needed for the article views data set, we could only process data from January to May 2012 within a reasonable time.

We use the edit history data dump of 1 June 2012 for the English Wikipedia, and 3 June for the German Wikipedia. Table 5.1 summarises the number of articles and revisions, and distinct usernames. Content articles are strictly encyclopedic articles and do not include articles for redirects, talk, user talk, help, and other auxiliary article types. We provide count of usernames and IP addresses in Table 5.1 to give indication of activity in the two Wikipedias.

The raw article views data set contains all of MediaWiki projects (including Wikipedia). We filter only revisions made in this time period from the edit history data. Table 5.2 provides some basic statistics on the raw data set filtered to view counts of English and German articles. Accordingly, we filtered the edit history data set to revisions made between January and May 2012. Despite this relatively small window of time, we show the feasibility and success of detecting vandalism using patterns of access within the article views data set.

[1] http://dumps.wikimedia.org/backup-index.html
[2] http://dumps.wikimedia.org/other/pagecounts-raw/

Table 5.1: Statistics of edit history data set. All revisions until start of June 2012.

Language	Content articles	Article revisions	Distinct usernames	Distinct IP addresses
English	4,000,264	305,821,091	4,020,470	25,669,884
German	1,419,217	65,732,032	447,603	5,565,475

Table 5.2: Statistics of article views data set. From January 2012 to May 2012.

Language	Articles viewed	Total views
English	2,261,593	4,567,904,954
German	805,964	1,493,732,111

5.2.1 Revisions containing Vandalism

From the raw revision data, every revision is reduced to a vector of features described in Table 5.3. These features are selected for their language independence and simplicity. For each revision, we analyse its comment for keywords for indication of a repair of vandalism (see Section 2.5.2), then we mark previous revision(s) to contain vandalism until the revision the editor reverted to.

To align the timestamp of revisions to the corresponding article views data set, we round up the revision time to the next hour. This ensures that the hourly article views references the correct revision when combining the two data sets. The alignment is performed on all revisions and should not affect classification.

We emphasise that user labelling of Wikipedia vandalism is noisy and incomplete. Some research provides solutions to this problem such as active learning [Chin et al., 2010], but a fully automated approach have inherent limitations as human involvement is necessary for some cases of vandalism [Wu et al., 2010]. We find about 2% of revisions between January to May 2012 contain vandalism. This is consistent with studies looking at these keywords [Kittur et al., 2007], but less than the 4-7% reported in other studies looking at vandalism beyond user labelling [Priedhorsky et al., 2007; Adler et al., 2010; Potthast, 2010].

5.2.2 Article Views

The raw article views data set is structured by views of article aggregated by hour. We perform a simple transformation and filtering of articles seen in the revisions data set above. The resulting features are summarised in Table 5.4. We also extract the redirect articles from the revisions data set and change all access to redirect articles to the canonical article. These extra view counts are aggregated accordingly.

These article views are important to seeing the impact of vandalism on Wikipedia [Priedhorsky et al., 2007]. The behaviour of vandals may also be seen in a change in access patterns, which may be from vandals checking on their work, or that article drawing attention from readers and their peers. A previous research study [Priedhorsky et al., 2007] (before the release of this data set) derived article views from the full Wikipedia server logs. This provides a much finer time unit for

Table 5.3: Feature description of edit history data set.

Attribute	Description
Article title	Unique identifier of a Wikipedia article.
Hour timestamp	The timestamp of this revision. In the format of YYYYMMDD-HH0000. The minutes and seconds are used to round up to the next hour.
Anonymous edit	The editor of this revision is considered to be anonymous if an IP address is given. We set this value to 0 for an edit by a registered user, and 1 for an edit by an anonymous user.
Minor revision	An editor can signify that they have made only superficial changes such as spelling, grammar, and formatting corrections. These changes do not require review by other editors and cannot be part of a dispute[3]. We set this value to 0 for normal revision, and 1 for minor revision.
Size of comment (bytes)	The size of the given comment of this revision.
Size of article text (bytes)	The size of the complete article of this revision.
Vandalism	This revision is marked as vandalism by analysing the comment of the following revision(s). We set this value to 0 for not vandalism, and 1 for vandalism.

Table 5.4: Feature description of article views data set.

Attribute	Description
Project name	The name of the MediaWiki project, where we are interested in Wikipedia projects in English (“en”) and German (“de”).
Hour timestamp	In the format of YYYYMMDD-HH0000, where YYYY for year; MM for month; DD for day of the month; HH for 24-hour time (from 00 to 23); and minutes and seconds are not given.
Article title	The title of the Wikipedia article. Article may not exist as the data set is derived from Web server request logs.
Number of requests	The number of requests in that hour. Not unique visits by users.
Bytes transferred	The total number of bytes transferred from the requests.

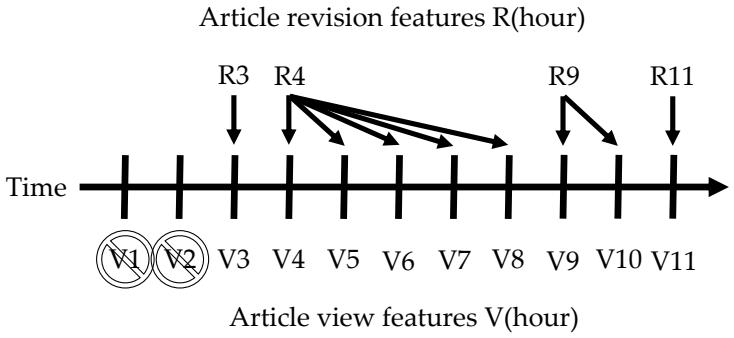

Figure 5.1: Illustration of the construction of the combined data set. Features from the article revision are added to the features from the article views. There are no article revision features for hour 1 (V1) and hour 2 (V2), so article view features for that hour are discarded.

analysis, but with a huge increase in data to process. With the time unit of hours, this data set provides coarse patterns of behaviours, but with manageable data size.

There are few research studies that use this data set. Most related research papers have developed tools for better access to this huge data resource and to provide simple graphs for topic comparison. One relevant study [Laurent and Vickers, 2009] use this data set to compare access to medical information on seasonal diseases like the flu. Access patterns in this data set reflect the oncoming of seasonal diseases. Wikipedia is accessed more than other online health information providers, and is a prominent source of online health information. Although vandalism is not covered, the seasonal access patterns allude to potential targets of vandalism.

To determine whether these article views occurred when articles are in a vandalised state, we scan the edit history data set and label all article views of observed vandalised or non-vandalised revisions. The unknown views from revisions made before January 2012, or articles without revisions in the 5 month period of study, are discarded. Thus, we have an article views data set labelled with whether the views are of vandalised revisions. The resulting size of the data is identical to the combined data set in the following subsection. This labelled article views data set allows us to determine whether view patterns can be used to predict vandalism.

From this resulting combined set, we split the "Hour timestamp" attribute into an "hour" attribute (see description in Table 5.4). This allows the machine learning algorithm to learn daily access patterns. In future work, we intend to experiment with monthly and yearly access patterns over a longer time period.

5.2.3 Combined Data Set

The combined data set is the result of merging of two time series data sets for each language. The data set is constructed by adding features from the labelled revisions data set to the labelled article views data set by repeating features of the revisions.

[3] https://en.wikipedia.org/wiki/Help:Minor_edit

Table 5.5: Statistics of the various data sets with percentage of vandalism cases.

Data set	Total number of articles	Total views of all articles	Combined (train)	Combined (test)
English	17,159,583	525,382,429	271,584,092	99,611,391
(vandal)	(2.08%) 356,618	-	(2.34%) 6,367,602	(2.04%) 2,033,838
German	3,731,714	284,932,083	139,967,644	55,010,679
(vandal)	(0.10%) 3,889	-	(0.06%) 86,534	(0.07%) 40,143

Thus for every article view, we have information on whether a vandalised revision was viewed and what the properties of that revision are. This merging process is illustrated in Figure 5.1.

We use the "hour" attribute split from the timestamp in the article views data set. Thus, we have the following 8 features in our combined data set: **hour, number of requests, bytes transferred, anonymous edit, minor revision, size of comment, size of article,** and **vandalism** (class label).

These features are language independent and capture the metadata of revisions commonly used, and access patterns. Note that we remove the article name as they are not necessary in evaluating the quality of classification. For example, access patterns of vandalised articles may be similar to other vandalised articles, regardless of the name of articles. For future work, we may identify the articles classified and further analyse to determine genuine cases of vandalism unlabelled or overlooked by editors.

To apply the classification algorithms, we split the combined data set by date into a training set (January to April) and a test set (May). The statistics of the data sets in this section are shown in Table 5.5 for comparison.

5.3 Cross-Language Vandalism Detection

We use the Scikit-learn toolkit [Pedregosa et al., 2011], which provides many well-known machine learning algorithms for science and engineering. We selected the following supervised machine learning algorithms from the toolkit:

- Decision Tree[4] (DT) – a non-parametric supervised learning algorithm that creates a classification model using decision rules derived from data features. We use the default Gini impurity criterion for determining the best split on a data feature. The toolkit implements an optimised version of the CART (Classification and Regression Trees) algorithm [Breiman et al., 1984], which is similar to the C4.5 algorithm [Quinlan, 1993].

- Random Forest[5] (RF) – a supervised ensemble classification algorithm that builds a model from many decisions trees trained on a sample drawn with

[4] http://scikit-learn.org/stable/modules/tree.html
[5] http://scikit-learn.org/stable/modules/ensemble.html#random-forests

replacement from the training data set. We use the same criterion as the DT classifier. The toolkit implements a variation of Breiman [2001], which averages the probabilistic prediction across many random trees (instead of having each tree vote for a class) to reduce the variance of the resulting model (compensating for increase bias compared to DTs).

- Gradient Tree Boosting[6] (GTB) – a supervised ensemble tree classification algorithm based on boosting to create an overall better classifier by optimising on a loss function. The sequential nature of creating a boosting model means GTB cannot be parallelised, in contrast to the RF classifier. We use the binomial deviance as the loss function because we have only two classes for the classification task. The toolkit follows the implementation of Friedman [2001] with input from other sources.

- Stochastic Gradient Descent[7] (SGD) – a simple and efficient approach for creating a linear classifier. SGD is highly scalable, but requires a lot of tuning of parameters, where the choice of loss function influences classification performance. After experimentation with small data sets, we use logistic regression as the loss function. The toolkit follows the implementation of Bottou and Bousquet [2008] and following work[8], with input from other sources.

- Nearest Neighbour[9] (NN) – a non-parametric classification algorithm . We use the KDTree data structure [Bentley, 1975] for its efficiency in determining points that are distant from each other, thus avoiding the brute-force search of a naive NN algorithm. The toolkit implements the *k*-NN algorithm [Altman, 1992], which determines the *k* (an integer) nearest neighbours for each query point.

We experimented with different settings available for the classifiers above, but we found there is little to no variance in the results. This is likely because all classifiers converged with the already large number of observations given.

From Table 5.5, we see the data set is highly unbalanced, which is unsuitable for some of our classifiers. We resolved this problem by undersampling the non-vandalism observations to match the number of vandalism observations. We apply this to all three data sets from Section 5.2: vandalised articles, article views, and combined. Thus, we built a balanced subset of the training and testing data.

We repeated the application of the classifiers to the balanced data to observe any effects from the random samples of non-vandalism observations. We found all classifiers seem to have converged with the already large number of observations in the balanced subset.

We also tried to train a Support Vector Machine (SVM) classifier, but we are unable to obtain results because of the different order in magnitude of training time.

[6] http://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting
[7] http://scikit-learn.org/stable/modules/sgd.html
[8] http://leon.bottou.org/projects/sgd
[9] http://scikit-learn.org/stable/modules/neighbors.html

Table 5.6: Approximate execution time of classifiers in seconds.

Time Taken (s)	DT	RF	GTB	SGD	NN
Training (en)	750	500	1800	5	20
Training (de)	3	4	15	1	1
Testing (en-en)	5	16	3	0.5	150
Testing (de-de)	0.5	0.5	0.5	0.5	2
Testing (de-en)	2	7	5	2	90
Testing (en-de)	0.5	0.5	0.5	0.5	4

We experimented with very few number of samples (0.1-1% of the data set) to obtain results for the SVM classifier within a reasonable time frame. However, we found all classifiers above and including the SVM performed poorly with the small number of observations.

For cross-language vandalism detection (see Section 2.6.1), we first train classification models for our two languages: English and German. These models are then evaluated on the testing set for the same language, and the testing set of the other language. The similarities between these language domains are captured by the language independent metadata features of Wikipedia. This cross-language application of models allows a generalisation of editing and viewing behaviour across Wikipedia.

This cross-language application of models has seen successful applications in the research area of cross-language text categorisation [Rigutini et al., 2005; Liu et al., 2012]. When considering text, cultural knowledge of the target language is needed to inform classifiers. The advantage of cross-language application of models is that one model can be used for multiple languages, saving resources developing models for each language. This is particularly relevant to Wikipedia with its large range of languages. Our research allows the potential generalisation of the concentration of vandalism research in English to other languages without additional inputs.

5.4 Experimental Results

Figures 5.2 to 5.7 present our classification scores for the five classifiers. These figures show the differences in AUC-PR and AUC-ROC scores for classification within language and out of language. For example, “en-de” means the classification model is trained on the English training set, then applied to the German testing set. The approximate execution times, gathered and averaged from multiple runs, are summarised in Table 5.6. Note that the execution times do not show the full parallelisation advantage of the RF model in training and testing models, which becomes significant in future chapters where the number of features and size of data sets are comparatively larger and more complex than presented in this chapter.

For the monolingual application of classification models in the single data sets (Figures 5.2, 5.3, 5.4, and 5.5), the tree based methods generally have better performance across both the AUC-PR and AUC-ROC scores. In particular GTB and RF

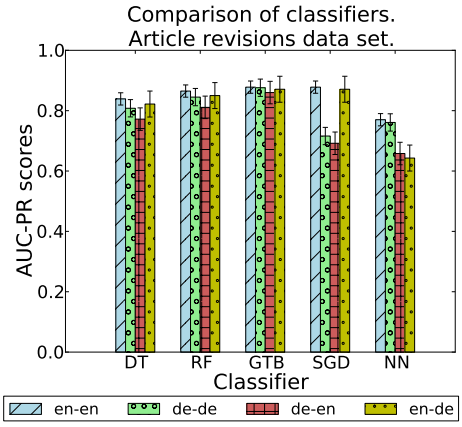

Figure 5.2: AUC-PR scores of classifiers for the article revisions data set. Key language1-language2 means classifiers are train in language1 and then applied to the testing set of language 2. Error bars show one standard error.

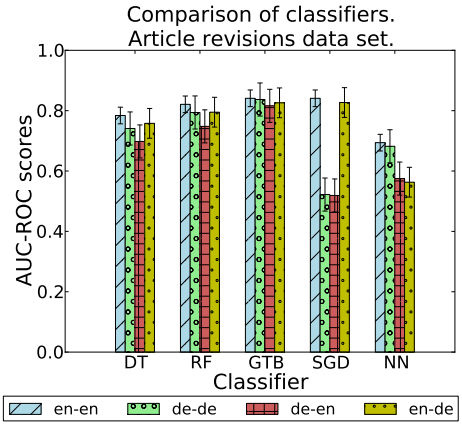

Figure 5.3: AUC-ROC scores of classifiers for the article revisions data set. Key language1-language2 means classifiers are train in language1 and then applied to the testing set of language 2. Error bars show one standard error.

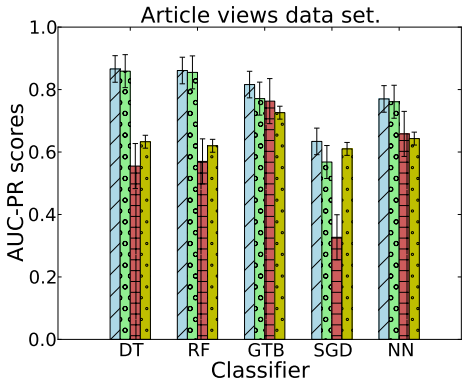

Figure 5.4: AUC-PR scores of classifiers for the article views data set.

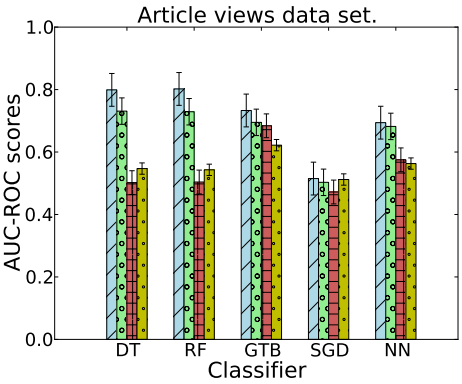

Figure 5.5: AUC-ROC scores of classifiers for the article views data set.

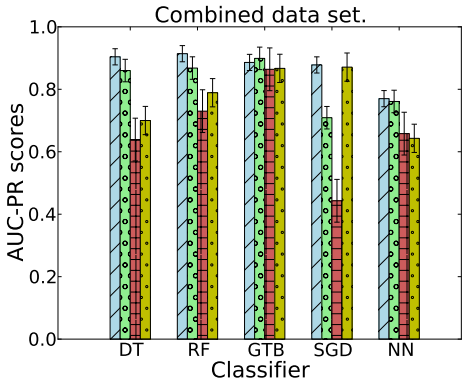

Figure 5.6: AUC-PR scores of classifiers for the combined data set.

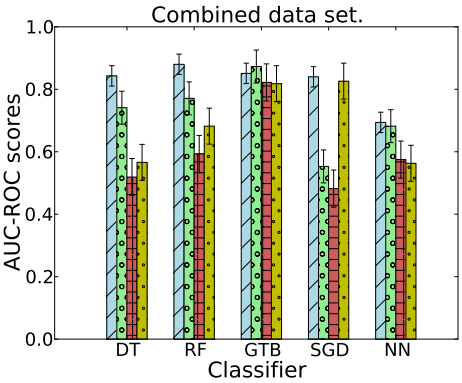

Figure 5.7: AUC-ROC scores of classifiers for the combined data set.

for the revisions data set, and DT for the views data set. However, they are also the most expensive models to train, with GTB taking the most time on average (see Table 5.6). These tree based methods have consistent high classification scores in the article revisions data set.

The cross-language application showed similar, but generally weaker, performance across all measures. GTB and RF continue to show generally better performance than the other classifiers. Interestingly, SGD performed best in the monolingual and cross-language cases when trained on the English revisions data (Figures 5.2 and 5.3), suggesting English may offer more patterns to detect vandalism. This is encouraging because SGD is the fastest algorithm to train. The cross-language application of models is not detrimental in most cases for all data sets, but with similar performance to the monolingual case. This suggests cross-language classification of vandalism is feasible with a variety of data sets.

In the combined data set (Figures 5.6 and 5.7), we see improvements to the classification scores, but mainly in the monolingual case. GTB continues to show high performance with improvements from the additional features. In general the combination of the data sets does not provide a significant advantage to the classifiers. The classifiers also seem to do well on the combined data set compared to individual data sets, but not much better. This suggests the classifiers are learning the best models from each data set, but improvements are not common.

The monolingual classification scores of the revisions data set in Figures 5.2 and 5.3 are comparable and better than many state-of-the-art systems. Note that the data sets used in various research studies are often constructed differently, and so care is needed when comparing different studies. From overviews of the PAN Wikipedia Vandalism Detection competition [Potthast et al., 2011, 2010], our results show better performance than many of entries, while using fewer features. The competition showcased multilingual entries in 2011, but no cross-language application of models is seen. White and Maessen [2010] presents an entry into the 2010 PAN vandalism competition and collated results from other Wikipedia vandalism research. We find our results for monolingual classification to generally have higher or similar AUC-PR scores.

Overall, we find the RF classifier to be most suitable for the cross-language vandalism detection task because of its fast training time and reasonable testing time, and also its similar high AUC-PR and AUC-ROC scores compared to the most robust but significantly longer training times of the GTB classifier. Furthermore, the RF classifier can be parallelised in contrast to the GTB classifier based on boosting (see descriptions of classifiers in Section 5.3); scalability is essential for learning vandalism models on the full Wikipedia data sets.

5.5 Discussion

Vandalism is an important cross-language issue on Wikipedia as more people contribute to and use Wikipedia as a resource in many different languages. The current

research on vandalism shows promising technologies to automatically detect and repair vandalism. However, these research studies largely concentrate on the English Wikipedia. The generalisation of these studies to other languages may not always be possible because of the independence of language domains, and the peculiarities in languages. Multilingual vandalism research is appearing, aided by construction of multilingual vandalism data sets, such as those by the PAN workshop. The cross-language vandalism detectors are ideal as models develop in one language can be applied to other languages.

In this chapter, we demonstrate the selection of language independent features allows cross-language vandalism classification, which have similar performance to a monolingual setting. Our results show the patterns in editing and viewing behaviour of editors and readers do generalise across language domains. This implies vandalism behaviour as gathered from these data sets is not culturally dependent.

Detecting vandalism using machine learning can be affected by the data and the features they present. The revisions data set is commonly used to extract features about vandalism. We show the article views data can also be used to predict vandalism, a novel way of using the data set. Access patterns from article views may indicate vandalism by showing odd patterns of interest, indicative of vandals observing their work, or odd increases in interest as more people find the vandalism and inform others.

Vandalism on Wikipedia would be apparent in all languages as the community of editors and number of articles grows larger. The success of cross-language vandalism detection would show the concentration of vandalism research in one language (English) is generalisable to other languages. Our data sets capture editing and viewing patterns, which characterise behaviour of vandals.

The advantages of the presented data sets are the simple to extract language independent features. These few features with the application of baseline classification algorithms outperform many past research studies. The combination of editing and viewing patterns shows some increase in performance, but generally allows classifiers to adapt to the best predictive features from both data sets individually. The article views data set may be too coarse to predict vandalism at the hourly level, but we found some classifiers can find patterns of vandalism as well, or better than the revisions data set in some cases. Furthermore, the features provided by the article views data set have strong contributions to the RF classifier as shown in Table 5.7, which details the information gain of the features in the combined data set. While the improvements are not reflected significantly in the results, the addition of the article view features should not be negatively impacting the performance of the classifiers compared to only using the article revision features.

We believe the RF classifier is the correct choice from our analysis in this chapter, and follows the de facto choice of related work in Chapter 3. The SGD classifier faced the same problem as the DT classifier in that it is not parallelisable, which is an important attribute for the later chapters where the number of features and input feature vectors are relatively larger and more complex. Furthermore, SGD had the slowest training time, which is significant compared to the other classifiers.

Table 5.7: Feature rankings of the combined data set as given by the Random Forest classifier. Bolded features are from the article views data set.

Combined (English)

Size of comment	0.358
Bytes transferred	0.261
Number of requests	0.196
Minor revision	0.086
Anonymous edit	0.042
Size of article	0.041
Hour	0.017

Combined (German)

Number of requests	0.354
Size of comment	0.243
Bytes transferred	0.238
Minor revision	0.089
Anonymous edit	0.036
Size of article	0.024
Hour	0.017

Some limitations of our approach include the rounding of hours, not using derived features, not analysing the content, and the necessity of the revisions data set to label the article views data set. The roll up of hours of revisions to match the article views data means that we have incomplete evidence of vandalism for each hour, so real-time decisions of vandalism cases may be incorrect until all evidence for that hour has been gathered. We can avoid this problem by obtaining article view statistics by the minute, which will result in vastly more data but allows for a smaller granularity to be analysed. The rich number of derived features used in other studies allows classifiers to learn more patterns of vandalism. This can often improve performance, but we find these data sets can be difficult to generate, especially when deploying solutions in bots. We have ignored the content of revisions, where word analysis may show the clear cases of unlabelled cases of vandalism. We show in the next chapter how content text features can be applied in the large scale required for Wikipedia and its many languages.

Overall, our data sets offer indications that vandalism can be detected with more complex techniques. The article views data set alone is not sufficient for vandalism detection and requires labelling from the revisions data set. However, the article views data set is a simple data set with few features that show some changes in access patterns when vandalism has occurred. We have shown cross-language application of vandalism models is feasible, and view patterns can be used to predict vandalism and may offer improvements to classifiers. In future work, we look to extracting more complex features from the article views data set that can show clearer changes in access patterns when articles are vandalised.

5.6 Summary

In this chapter, we have presented data sets for vandalism detection and demonstrated the application of various machine learning algorithms to detect vandalism within one language and across languages. We developed three data sets from the hourly article view count data set, complete edit history of Wikipedia, and their combination. We looked at two language editions of Wikipedia: English and German. We found the GTB classifier showed generally best performance in predicting vandalism,

despite being the most time consuming algorithm. The RF classifier provides strong classification performance similar to the GTB classifier, but with the fastest training time of the tree classifiers, making it the most suitable for cross-language vandalism detection. The RF classifier is also parallelisable, in contrast to GTB, which is limited by the sequential nature of the boosting algorithm. These results show the view and edit behaviour of vandals is similar across different languages. The implication of this result is that vandalism models can be trained in one language and applied to other languages.

In future work, we look to extend the time span of the data set and apply to other languages. This would provide further evidence for the general applicability of classification models cross-language to detect vandalism using this combined data set. We would like to continue with our data set and feature analyses to determine factors such as the effect of popular articles and traffic changes for vandalised articles. From these analyses, we could add further features and derive other features, such as ratios, to enrich the data set and also explore other data balancing techniques. We could improve the baseline classifiers by building classifiers more suited to this data set. In the long term, we plan to have this system able to generate the combined data set to identify possible cases of vandalism for closer analysis.

The next chapter continues our investigation into cross-language vandalism detection (CLVD) by developing novel text features from the text content of revisions. Analysing the text of articles is computationally intensive, but it is the most certain method of identifying vandalism and recovering evidence of vandalism.

Chapter 6

Text Features for Cross-Language Vandalism Detection

In this chapter, we show how feature engineering can be used to find suitable vandalism features for cross-language vandalism detection (CLVD). We propose novel text features that allow a machine learning classifier to better distinguish vandalism and vandalism repaired across languages compared to text features used in past research. These features also allow us to study the contributions of counter-vandalism bots, which are often ignored in related work. We show differences and contrast features important to bots and users in distinguishing vandalism on Wikipedia across five languages. In contrast to the metadata features of the previous chapter that are not representative of the text processing actions of bots, we can identify the important features bots look for in detecting vandalism without having to deconstruct and analyse each bot.

This chapter has been published in Tran and Christen [2014] and is organised as follows. Section 6.1 introduces the problem and motivates the solutions of this chapter. Section 6.2 highlights the additional data processing needed for this chapter from Section 2.5, and Section 6.3 details and ranks the language invariant text features of vandalism. Section 6.4 describes our cross-language learning method, and Section 6.5 summarises and compares our results to the PAN data sets. Section 6.6 compares our results to related work, and Section 6.7 discusses our findings, advantages, and limitations. Finally, Section 6.8 concludes this chapter and provides future directions for research.

6.1 Introduction

The use of counter-vandalism bots is changing the way Wikipedia identifies and bans vandals [Geiger and Ribes, 2010; Halfaker and Riedl, 2012]. However, contributions by bots are often not considered nor discussed, despite their importance to Wikipedia and some bots becoming the most prolific editors [Adler et al., 2008; Geiger and Ribes, 2010]. The increasing delegation of vandalism detection to bots poses interesting research questions: how do the detection rates of bots and users compare to each other, and how do they differ across different Wikipedia languages?

We investigate these questions by learning vandalism collectively recognised by bots and users, and evaluate these models against both bots and users across 500 million revisions from five different languages: English (en), German (de), Spanish (es), French (fr), and Russian (ru). We propose a new set of computationally efficient features that are language invariant, and have classification performance comparable to the previously proposed features. We show bots and users have similar vandalism identification scores when we apply their classification models on the other's recognised set of vandalism cases. Furthermore, we show that combinations of vandalism classification models generalise well across languages without statistically significant loss in classification quality. To strengthen our results, we replicate our experiments on the PAN vandalism data sets of approximately 62,000 revisions from competitions held for the PAN workshops [Potthast, 2010] (see Section 2.5), and discuss limitations with these data sets.

The contributions of this chapter are (1) developing novel text features that capture language invariant aspects of vandalism, and have greater effectiveness compared to features from related work as demonstrated by a statistical test and feature ranking; (2) contrasting the differences between bots and users by learning vandalism identified by bots and users; (3) demonstrating that cross-language application of classification models do not have significant loss in classification quality; (4) conducting our experiments on the entire Wikipedia data dumps (over 500 million revisions), which comprehensively includes all random samples of revisions in the PAN baseline data sets; and (5) replicating our experiments on these much smaller PAN baseline data sets, showing and contrasting the performance of features often used in related work on these data sets and on the full Wikipedia data dumps.

6.2 Wikipedia Data Sets

We use the first Wikipedia data dump available in 2013 of all revisions of every article from five languages: English (en), German (de), Spanish (es), French (fr), and Russian (ru). We further describe these data sets and the data processing steps in Section 2.5. We split these data sets into training sets (all revisions before 2012) and testing sets (all revisions in 2012). The testing sets contain between 9-30% of all revisions for each language. In this chapter, we distinguish the contributions of bots and users (human editors), compare and contrast the vandalism that a classifier can learn from their repairs of vandalism.

6.3 Feature Engineering

We generate our features from words extracted from the difference of the content of the repaired revision with the previous revision, which contains vandalism. From the diff[1] algorithm (process described in Section 2.5), we have lines (separated by periods) unique to the revision before the repair, lines unique to the revision after

[1] https://docs.python.org/2/library/difflib.html

the repair, and the lines changed in the repairing process. We ignore common lines to accurately determine changes in content. The common lines can show the ratio of the vandalised content to normal content, but for cases such as mass deletes, the size of lines unique to the repaired revision is sufficient to show this case. We further perform a sentence difference to extract vandal words that were repaired. Our text processing uses unicode (UTF-8) encoding and language specific alphabets.

All features are shown in Table 6.1 with a summarised description, an average time of generating features in milliseconds (ms), and a Kolmogorov-Smirnov (K-S) statistical test [Massey, 1951] (described in Section 6.3.4). We order our features in groups of relatedness, where bolded features are our novel contributions to detecting vandalism. Note that our features are applied specifically to diff words instead of the full diff of revisions as in previous works. Our borrowed features are text features from the winners of the PAN 2010 and 2011 workshops [Mola-Velasco, 2010; West and Lee, 2011], where they first appeared for the use of detecting vandalism.

Features `F00-NLB` to `F09-TWD` are generated from the revisions before and/or after a repair. Features `F10-PW` to `F20-SC` are generated from the words changed in the repair, which isolate possible vandal words and captures distributions of words in the repair. Note that duplicate words can exist and we count these in some features. Features `F21-UL` to `F31-WS` are applied on each word that was repaired, where we select for values that indicate vandalism. Although some features are derivatives from related work [Mola-Velasco, 2010; West and Lee, 2011], we justify their novelty by our application to lists of single words – further polarising vandalism cases – and show their effectiveness on the full Wikipedia data set.

6.3.1 Data Modification Features

Although these features are novel, they are intuitive in capturing the changes in content. We focus on changes reported by our diff algorithm.

Features `F00-NLB` to `F03-NLCA`: These features are a count of types of lines from the diff algorithm. High counts of unique lines in the vandalised revision (before the repair) indicate mass insertions, and high counts in the repaired revision (after repair) indicate mass deletions. The count of line changes indicates small changes that may show vandalised insertions or changes of text. Note that these features are similar to the byte change in West and Lee [2011], but we further polarise the impact of changes with these features.

Features `F04-DTLW` to `F09-TWD`: Similar to the line counts, we count the changes of words before and after a repair. These changes in the words of the repair show the subtler cases of vandalism that modify specific words. The difference of word lengths and number of words show the extreme changes needed to repair vandalism, whereas the ratios show the relative size of changes needed for repair. Similarly, the lengths and the counts of the unique words show the relative change in size and the absolute number of changes needed in repairing vandalism. These combinations ensure that we can identify the repairs made by bots and users of subtler vandalism.

Table 6.1: Features generated from the revision before (b) and/or after (a) a repair (`F00-NLB` to `F09-TWD`) and the words changed (`F04-DTLW` to `F09-TWD`), and the properties of words (`F10-PW` to `F31-WS`). Bold features are novel contributions. Detailed description of features is given in Section 6.3 and of the Kolmogorov-Smirnov (K-S) test in Section 6.3.4. Note that the timing is for generating each feature individually – not including the required diff – and does not reflect parallelisation and grouped preprocessing of required data.

Feature	Description	Time (ms)	Failed K-S (Full)	Failed K-S (PAN)
F00-NLB	Number of unique lines in (b)	0.035	10%	0%
F01-NLA	Number of unique lines in (a)	0.035	0%	50%
F02-NLCB	Number of unique lines changed in (b)	0.035	10%	50%
F03-NLCA	Number of unique lines changed in (a)	0.035	10%	50%
F04-DTLW	Difference of total lengths of unique words of (b) and (a)	0.400	0%	25%
F05-RTLW	Ratio of total lengths of unique words of (b) and (a)	0.400	10%	25%
F06-DTNW	Difference of total number of unique words of (b) and (a)	0.385	0%	0%
F07-RTNW	Ratio of total number of unique words of (b) and (a)	0.385	10%	25%
F08-NWD	Number of unique words	0.004	10%	0%
F09-TWD	Number of all words	0.003	10%	0%
F10-PW	Pronoun words	0.010	50%	100%
F11-VW	Vulgar words	0.007	50%	100%
F12-SW	Slang words	0.007	30%	50%
F13-CW	Capitalised words	0.006	10%	0%
F14-UW	Uppercase words	0.006	10%	75%
F15-DW	Digit words	0.004	20%	50%
F16-ABW	Alphabetic words	0.006	10%	0%
F17-ANW	Alphanumeric words	0.006	10%	0%
F18-SL	Single letters	0.007	20%	0%
F19-SD	Single digits	0.004	20%	75%
F20-SC	Single characters	0.005	80%	100%
F21-UL	Highest ratio of upper to lower case letters	0.170	0%	25%
F22-UA	Highest ratio of upper case to all letters	0.170	0%	25%
F23-DA	Highest ratio of digit to all letters	0.170	0%	25%
F24-NAN	Highest ratios of non-alphanumeric letters to all letters	0.170	0%	25%
F25-CD	Lowest character diversity	0.115	0%	25%
F26-LRC	Length of longest repeated character	0.175	10%	50%
F27-LZW	Lowest compression ratio, lzw compressor	3.800	0%	25%
F28-ZLIB	Lowest compression ratio, zlib compressor	0.275	10%	25%
F29-BZ2	Lowest compression ratio, bz2 compressor	0.475	0%	25%
F30-WL	Longest unique word	0.040	10%	25%
F31-WS	Sum of unique word lengths	0.040	10%	0%

6.3.2 PAN Workshop Features

We borrow these features directly from the winners of the PAN workshops, where they have been often used by related work (see Section 3.4.1). The features are adapted for our data sets where needed and we provide clearer sources for vulgar and slang words.

Features `F10-PW` to `F12-SW`: Three types of words common or indicative of vandalism are pronouns, slang, and vulgarity. We extract these words from Wiktionary[2] for each language, where available. For all languages considered, we have 105 pronouns, 8,465 slang words, and 2,250 vulgar words. We search for all these words in the sentence diff for all languages. For example, if English vulgarities are used in German vandalised revisions, these vulgar words are counted in the features for the German revisions. These features have previously been used in related work [Mola-Velasco, 2010; West and Lee, 2011], but for English only and with an unknown source of the vocabulary. Our visual inspection shows that vulgar and slang words are not likely to be benign words in other languages. Interestingly, some vulgar words from other languages are included in English.

Features `F13-CW` to `F20-SC`: We count the different word types. By looking at the letters of each word, some indications of possible vandalism are uppercase words, words with digits, and words that are single letters. These features are common indicators of vandalism in related work [Mola-Velasco, 2010; West and Lee, 2011].

6.3.3 Word Level Features

These novel features are modified from related work to suit our word level analysis. In a sentence difference, we expect a single oddity in a word to indicate vandalism, hence we do not aggregate or average values as a vandal can avoid detection by simply masking vandalism with unrelated but legitimate words.

Features `F21-UL` to `F25-CD`: These features look at the ratios of letters to words. We select these features with definitions from Mola-Velasco [2010], but apply them with modifications to the equations as need to suit the word level instead of the document level. We take the maximum or minimum of these ratios for each word as a strong indicator of vandalism.

Features `F26-LRC` to `F29-BZ2`: Feature `F26-LRC` shows the length of the longest repeated character in a word as used in Mola-Velasco [2010], which is often a clear case of vandalism. To complement this feature, the compressibility of words can identify abnormally long repeated sequence of letters. We compare three compression algorithms and take the lowest compression ratio, indicating the highest compressibility of a word. Features `F28-ZLIB` and `F29-BZ2` are provided to extend and contrast the compression feature `F27-LZW` from Mola-Velasco [2010]. These are the most computationally intensive features as they require compression, but we maintain a lookup table of compressed words to avoid repeated computation.

[2] http://www.wiktionary.org/

Features F30-WL to 31-WS: We count the longest unique words and the total size of the unique words in the sentence difference. These are intuitive features from Mola-Velasco [2010] and West and Lee [2011], but with a different interpretation and application.

6.3.4 Kolmogrov-Smirnov Statistical Test

We use the two-sample Kolmogorov-Smirnov (K-S) statistical test [Massey, 1951] from the SciPy toolkit[3] to determine whether the features distinguish the normal revisions from the vandal revisions – repaired by bots and users – at the 0.05 significance level. The K-S test provides an indicator of whether features may be beneficial to statistical machine learning algorithms. We have 10 data sets for the full Wikipedia (Full) data set (5 languages with bots and users for each language) and 4 data sets for the PAN data set (1 language for 2010, and 3 languages for 2011). We show the percentage of data sets failing the K-S test at the 0.05 significance level in Table 6.1.

We immediately see that our novel features are generally more effective in distinguishing normal revisions from vandal revisions from the repairs – with the lower percentage of failure, especially in the much larger full Wikipedia data sets. Some of the borrowed features from the PAN workshops (F10-PW to F20-SC) are not effective in the PAN data sets, and are less effective in the full Wikipedia data set. The small size of the PAN data sets may also hinder many other features that are effective in distinguishing vandalism in the full Wikipedia data sets. For example, the size of the lines changed (F02-NLCB and F03-NLCA), and words with many repeated characters (F26-LRC).

The higher failure of K-S tests may be explained by the PAN data sets containing more difficult or ambiguous cases of vandalism that require manual analysis. This means the features may be capturing specific types of vandalism that are abundant in the full Wikipedia data sets but not the PAN data sets because of different vandalism selection methods. The K-S test only provides an indicator of the effectiveness of features as different features may show strong performance in one language but not any others. Thus, we advocate for evaluation of features on both the PAN data sets and the full Wikipedia data sets, as we have done in this chapter, but also care in using the K-S measure by considering the effectiveness of features in each language and not as an indicative performance in aggregate.

6.3.5 Feature Ranking

We use the Random Forest classifier from the Python based Scikit-learn toolkit [Pedregosa et al., 2011] to rank these 32 features by their importance. This is further statistical evidence showing the general effectiveness of our feature sets before use in classification. Table 6.2 shows the top 5 features ranked by their information entropy (IE) scores (as used by the Random Forest classifier) for each language and for bots and users. The scores show the features that give the most homogeneous branches

[3] http://docs.scipy.org/

Table 6.2: Top 5 features as determined by the Random Forest classifier (Section 6.4). We show in bold features that are our contribution. Scores are the information entropy (IE).

Wiki	en		de		es	
Type	Feature	IE Score	Feature	IE Score	Feature	IE Score
Bots	**F01-NLA**	0.012	**F01-NLA**	0.016	**F01-NLA**	0.016
	F12-SW	0.009	**F00-NLB**	0.010	**F24-NAN**	0.011
	F00-NLB	0.008	**F24-NAN**	0.008	**F07-RTNW**	0.009
	F04-DTLW	0.007	**F05-RTLW**	0.007	F11-VW	0.006
	F07-RTNW	0.006	F17-ANW	0.006	**F04-DTLW**	0.005
Users	**F00-NLB**	0.010	**F05-RTLW**	0.011	**F04-DTLW**	0.011
	F04-DTLW	0.009	**F04-DTLW**	0.011	**F05-RTLW**	0.009
	F05-RTLW	0.008	**F07-RTNW**	0.008	**F00-NLB**	0.008
	F07-RTNW	0.007	**F06-DTNW**	0.007	**F07-RTNW**	0.007
	F06-DTNW	0.006	**F01-NLA**	0.006	**F06-DTNW**	0.005

Wiki	fr		ru	
Type	Feature	IE Score	Feature	IE Score
Bots	**F04-DTLW**	0.013	**F24-NAN**	0.011
	F01-NLA	0.012	**F01-NLA**	0.010
	F06-DTNW	0.011	**F30-WL**	0.010
	F00-NLB	0.008	**F23-DA**	0.008
	F11-VW	0.007	**F21-UL**	0.008
Users	**F05-RTLW**	0.012	**F04-DTLW**	0.009
	F04-DTLW	0.009	**F05-RTLW**	0.008
	F01-NLA	0.007	**F00-NLB**	0.006
	F00-NLB	0.007	**F07-RTNW**	0.006
	F06-DTNW	0.007	**F31-WS**	0.006

in the forest of decision trees (i.e. the amount of information gained after splitting on that feature in a decision tree). For example, for bots in the English Wikipedia, we gain twice as much information when splitting on feature `F01-NLA` (0.012) than on feature `F07-RTNW` (0.006), while for users the differences in the top five features are less. The IE scores are an average of 10 training iterations of the classifier to account for the randomness in the Random Forest classifier.

For bots, we find some of our new features are consistently important for most languages. For example, features `F01-NLA` and `F00-NLB` both show cases of mass deletions and insertions, respectively. Feature `F24-NAN` is important for German, Spanish, and Russian Wikipedias, indicating high uses of non-alphanumeric characters in vandal words. Features `F04-DTLW` and `F07-RTNW` – important for the English and Spanish Wikipedias – show the total difference and ratio of lengths of words before to after the repair, which indicates many insertions of vandal words in sentences and insertion of long words in the case of the French Wikipedia. Interestingly, slang words (`F12-SW`) is one of the most important features in the English Wikipedia, indicating frequent use in vandalism cases. In general, bots identify vandalism features

that show changes in text and word sizes, and changes that introduce vulgar or slang words.

For users, we see a common set of important features across most languages, namely the word modification features `F04-DTLW` to `F07-RTNW`, and in particular `F05-RTLW` for all languages. Feature `F05-RTLW` suggests the vandal words are disproportionate in ratio size to the repaired words. These features – `F04-DTLW` to `F07-RTNW` – suggest vandal words are out-of-place with respect to the sentence they were in and these types of potentially subtle vandalism are consistently being identified by users across all languages.

Overall, there are differences in the importance of features for bots and users. Bots seem to handle more prominent vandalism features such as mass insertions and deletions of text, and slang and vulgar words. Features important to users are based on the changes made and the length of words used in the vandalised revisions. This suggests users are repairing subtle vandalism that requires deep inspection of words.

6.4 Cross-Language Learning

We split the Wikipedia data sets into training (all revisions before the year 2012) and testing (all revisions in the year 2012). The data set is highly imbalanced, so we undersample (without replacement) the normal revisions to match the number of identified vandalised revisions for the training and testing sets. This allows the Random Forest algorithm to improve its classification performance with many balanced tree samples. We address the issue of training data balancing in Section 6.5.5, where we compare other ratios of normal revisions to vandalised revisions to show there are no statistically significant changes in classification results for different sampling ratios. We perform cross-language learning and detection as described in Section 2.6.1.

6.5 Classification Results

We use the Random Forest classifier and evaluation metrics from the Python based Scikit-learn toolkit [Pedregosa et al., 2011]. This classifier was shown to be the most robust and generally best performing classifier from related work on vandalism detection [Adler et al., 2011] and in CLVD as shown in Chapter 5, hence we did not compare different classifiers in this chapter. To maximise performance, we conduct a grid search with 10-fold cross validation on the training data over a wide range of the classifier parameters for each language, such as the number of estimators (trees in the forest), maximum number of features, minimum number of samples per leaf, minimum number of samples for split, and minimum density. We present our classification results as the area under the precision-recall curve (AUC-PR), but not the area under the receiver operating characteristic curve (AUC-ROC) because of the already numerous AUC-PR results for combinations of languages. In Section 2.7, we describe these measures and make an argument for favouring AUC-PR scores as they show whether vandalism cases are being correctly identified.

Table 6.3: Classification results of all features in Section 6.3 extracted from the PAN 2011 vandalism data set.

AUC-PR	Test		
Train	en	de	es
en	**0.768**	0.715	0.774
de	0.691	**0.744**	0.731
es	0.756	0.703	**0.789**
all	**0.771**	0.729	**0.803**

6.5.1 Baseline Comparison: PAN Data Sets

Previous Wikipedia vandalism detection studies have focused mainly on the PAN data sets as described in Section 2.5 and Section 3.4.1. We use the PAN data sets as a baseline comparison of results by evaluating our features under the same conditions as the full Wikipedia data set, with a 1:1 ratio of classes. We also apply cross-language learning on the PAN 2011 data set (as far as we are aware, we are the first to do so).

The PAN 2010 baseline data set contains 32,440 revisions sampled from the English Wikipedia, with approximately 7% vandalised cases. At the 50% random sampled split of the data into training and testing sets, which is reflective of the competition at the time, we have an AUC-PR score of 0.768.

The PAN 2011 vandalism baseline data set contains a total of 29,952 revisions sampled from each of the English, German, and Spanish Wikipedias. A total of approximately 9.4% are vandalised revisions. With a similar 50% random sampled split, we have classification scores as shown in Table 6.3.

Some limitations with the PAN data sets are unrepresentative samples of bots (described in Section 3.4.1) despite counter-vandalism bots having a strong presence on Wikipedia since 2006 [Geiger and Ribes, 2010; Adler et al., 2008] – especially in the English Wikipedia, and the potential bias with sampling from 'important' articles [Potthast, 2010]. However, the value of the PAN data sets comes from the manual evaluation, which may contain very difficult or ambiguous vandal edits that can only be identified by consensus.

We believe this is the reason for the comparatively lower AUC-PR scores for the PAN data sets compared to our results in Tables 6.4 and 6.7 for all matching pairs of training and testing languages. However, for the full Wikipedia data sets, our features have strong classification performance within and across languages.

Overall, many features presented in related work show strong classification performance on the PAN data sets, but we believe they also need to be evaluated on the full Wikipedia data set to gauge their effectiveness in distinguishing vandalism within and across languages on large scale data. Each type of data set have its advantages and disadvantages that are complementary, such as manual verification of vandalism versus automated gathering from comments, and the relative sizes of the data sets, as discussed in Section 2.5.

Table 6.4: Results of cross-language and cross-user type classification for the Random Forest (RF) classifier. Bold entries are the same match ups of language (diagonal) and user type, and the highest score in each column.

AUC-PR	Test	en		de		es		fr		ru	
Train	Type	bots	users	bots	users	bots	users	bots	users	bots	users
en	bots	**0.956**	0.797	0.946	0.734	0.943	0.778	0.870	0.798	**0.750**	0.743
	users	0.937	**0.814**	0.936	0.743	0.929	0.787	0.849	0.812	0.432	0.759
de	bots	0.917	0.777	**0.933**	0.730	0.914	0.776	0.814	0.781	0.432	0.742
	users	0.914	0.800	0.918	**0.749**	0.922	0.783	0.808	0.806	0.597	0.759
es	bots	0.929	0.777	0.945	0.721	**0.950**	0.768	**0.881**	0.787	**0.750**	0.732
	users	0.911	0.792	0.922	0.741	0.935	**0.790**	0.847	0.800	0.432	0.760
fr	bots	0.936	0.772	**0.950**	0.738	0.939	0.776	**0.864**	0.780	**0.750**	0.738
	users	0.904	0.801	0.917	0.742	0.921	0.783	0.824	**0.817**	0.615	0.761
ru	bots	0.754	0.700	0.788	0.678	0.775	0.715	0.702	0.712	**0.513**	0.711
	users	0.861	0.753	0.896	0.729	0.881	0.757	0.757	0.767	0.531	**0.778**

6.5.2 Combinations of Classification Languages

For the full Wikipedia data sets, the results of combinations of training and testing data are presented in Table 6.4. The rows of the table are the language and user type data set a classifier is trained on, and similarly the columns show testing set for the classifier. We show in bold results of the same language and the same user type of the training and testing set, and also the highest scores of each column.

The Russian (ru) training and testing sets for bots are relatively small compared to other languages, as seen in Table 2.4. These few vandalism observations generally result in poor classification performance from all languages for the Russian bots training and testing sets. However, the training set provides many common patterns for those few observations, where performance is poor compared to the training sets of other languages. The relatively large number of vandalism cases in the Russian training set for users show higher classification performance on other languages.

Within the same language and user type (diagonal bold entries), the classifier shows some of the highest scores amongst the language combinations. The exceptions are scores of the German and French bots, where the classifier trained on data of the English bots show better classification performance. This suggests English bots can identify more vandalism cases identified by bots in the German and French Wikipedias than the German and French bots.

For bots in each language, we find they have generally high classification performance on vandalism identified by bots from another language. This suggests bots have consistent behaviour, so there is little variation in the way they identify vandalism. When we applied these models to users in different languages, we find lower classification performance. This suggests users are identifying a wider range of vandalism types than bots, which is expected.

For users in each language, we find consistent high performance on vandalism

Table 6.5: Student's t-test p-values calculated from Table 6.4 (diagonally symmetric matrix of combinations of training and testing sets). We bold values less than the 0.05 level.

p-value	Table row	en		de		es		fr		ru	
Table row	Type	bots	users	bots	users	bots	users	bots	users	bots	users
en	bots	–	0.35	0.13	0.15	0.08	0.26	0.06	0.16	**0.00**	**0.02**
	users		–	**0.00**	0.75	0.48	0.08	0.48	0.67	**0.00**	0.13
de	bots			–	0.17	0.21	**0.02**	0.20	0.17	**0.01**	0.50
	users				–	0.33	0.49	0.31	0.35	**0.00**	**0.00**
es	bots					–	0.36	0.94	0.37	**0.00**	0.05
	users						–	0.36	0.43	**0.00**	0.21
fr	bots							–	0.35	**0.00**	**0.04**
	users								–	**0.00**	**0.00**
ru	bots									–	**0.00**
	users										–

identified by bots for most languages. This suggests users look for similar patterns of vandalism as bots. The numerous users in the English Wikipedia identify a higher portion of vandalism across languages than users from other languages. This suggests with more users (human editors), more vandalism patterns can be identified.

For each row to each other row of results, we apply the t-test to find if there are any statistically significant differences in performance when learning across languages and editor types. In Table 6.5, we show an example calculation of p-values for all pairs of language and user combinations for Table 6.4. Each value is calculated from a row of Table 6.4 with the other rows. For example, p-value 0.35 is calculated from the "en bots" row and "en users" row of Table 6.4. We use the Excel TTEST function to calculate a two-tailed paired t-test of the rows. We do not include the full matrix of paired t-test p-values for other tables for brevity.

We find that in general learning on any language and any editor type does not show significant differences in classification performance across all languages and both editor types, at the 0.05 level. There a few exceptions, but mainly when learning from the Russian bots, because of the notably fewer number of training samples for Russian bots. The t-test p-values on rows suggest vandalism can be learned from any of the presented training sets (except Russian bots) and applied to other languages and editor types without significant differences in classification quality.

When looking specifically at the testing data of users for each language (users columns), we find there is a difference in classification quality between the row of bots and users for each language, with many t-test values less than the 0.05 level. This suggests that there is a difference in how bots and users recognise the vandalism identified by users across languages. However, we do not see this difference between bots and users for the vandalism identified by bots (using the bots columns). This suggests users identify a wider range of vandalism that includes vandalism that bots can identify.

Table 6.6: Results of cross-language and cross-user type classification for the Random Forest (RF) classifier. The training data for bots and users are combined for each language, and all training data for bots and users are combined. Bold entries are the same match ups of language (diagonal) and the highest score in each column. Statistical significance of results are discussed in Section 6.7.

AUC-PR	Test	en		de		es		fr		ru	
Train	Type	bots	users	bots	users	bots	users	bots	users	bots	users
en	both	**0.954**	**0.816**	0.938	0.751	0.936	0.789	**0.882**	0.815	**0.750**	0.756
de	both	0.916	0.801	**0.926**	**0.752**	0.923	0.782	0.827	0.808	0.597	0.757
es	both	0.934	0.791	0.947	0.737	**0.955**	**0.788**	0.877	0.802	**0.750**	0.747
fr	both	0.925	0.799	0.927	0.742	0.939	0.786	**0.840**	**0.817**	**0.750**	0.757
ru	both	0.857	0.749	0.877	0.717	0.863	0.757	0.755	0.766	**0.481**	**0.776**
all	bots	**0.956**	0.797	**0.953**	0.740	0.947	0.783	0.875	0.801	**0.750**	0.746
all	users	0.938	0.815	0.933	0.752	0.933	0.790	0.850	0.815	0.432	0.771
all	both	0.952	**0.816**	0.948	**0.755**	0.941	**0.793**	0.867	**0.818**	**0.750**	0.770

6.5.3 Combined Training Data

As a further investigation, we combine the training data of bots and users for each language, for each editor type, and for all languages and both editor types. These classification results are presented in Table 6.6. This experiment investigates the common practice of learning vandalism without distinguishing contributions of bots and users.

By learning from both bots and users for each language, we find some differences in classification performance. Related works do not make this distinction, which can result in higher classification scores because of the predictability of bots in detecting specific types of vandalism. Bots follow rules and common structures of vandalism, which machine learning algorithms can recognise quickly, leading to more correct results and so a higher AUC-PR. In contrast, human detected cases of vandalism have a greater variance in the types as these are the vandalism cases bots may not recognise.

For the same language of training and the same editor types for testing (bold diagonal language entries of Table 6.4 and Table 6.6), we only find t-test p-values greater than the 0.05 level. Thus, there is no statistically significant difference when learning vandalism from both bots and users or each individually.

Similarly to the previous subsection, we find no statistically significant difference when comparing the rows of Table 6.6 to rows of Table 6.4, with the exception of Russian bots. Combining training data from all languages from bots or users, and both, we also find no differences at the 0.05 level. This shows there is no difference in learning vandalism from bots and users across all languages considered.

We also observe the same statistically significant difference when looking specifically at the testing data of users for each language (users columns); and the same non-difference of the testing data of bots for each language (bots columns). So, combining observations from bots with users may not improve detection performance

Table 6.7: Results of cross-language and combined editor types. Bold entries are the same match ups of languages, and the highest score in each column.

AUC-PR	Test				
Train	en	de	es	fr	ru
en	**0.895**	0.767	0.858	0.815	0.756
de	0.867	**0.766**	0.848	0.808	0.757
es	0.873	0.754	**0.864**	0.803	0.747
fr	0.871	0.757	0.856	**0.817**	0.757
ru	0.812	0.729	0.805	0.766	**0.775**
bots	0.886	0.757	0.857	0.801	0.745
users	0.886	0.768	0.857	0.816	0.771
all	0.894	**0.771**	0.861	**0.819**	0.769

for vandalism identified by users. This suggests users do identify a wider range of vandalism, where the contributions of bots may not be different across languages, but can provide some improvements to classification performance. Although these improvements may seem relatively small, this can mean thousands more cases of vandalism are automatically detected everyday across many languages.

6.5.4 Combined Training and Testing Data

To complete the cross-language learning and have data comparable to related work, we combine the editor types for the training and testing data. Table 6.7 presents cross-language classification results for each language and combined training for bots and users of all languages, and all training data.

Results of the matching training and testing languages show AUC-PR in-between those of bots and users in Tables 6.4 and 6.6. Similarly, the t-test p-values of the rows show values greater than 0.05, except when combining the training data of all languages. The last row of Table 6.7 shows t-test p-values less than 0.05 for most of the other rows. This suggests by using all training data, we have a statistically significantly better vandalism detector for all languages.

Our results are consistent for the testing language, suggesting related languages, such as English and German, and Spanish and French, do not affect the classification results. This is further evidence for the language independent nature (for the languages considered) of the proposed features.

6.5.5 Effects of Data Sampling

We sampled (in Section 6.4) the over-represented normal revisions from the Random Forest classifier because this allows more balanced decision trees to be built in the classifier to distinguish vandalism, reduces the size of the models and data needed for training, and reduces learning time. However, data sampling raises questions about bias in performance. We present the 1:1 (one to one) ratio of normal revisions

to vandalised revisions in Table 6.7, but we have repeated our experiments for the sampling ratios of 2:1, 4:1, 10:1, and 13:1. The ratios of 10:1 and 13:1 represent the approximate ratio of vandalised revisions observed in the PAN 2011 and PAN 2010 data sets, respectively.

We also present our results for training on the balanced data set 1:1 (tr), and applying to the unbalanced testing sets 10:1 (te) and 13:1 (te). These results simulate the real-world effects of learning on a balanced data set and applying to a non-balanced data set, such as in the full Wikipedia corpus.

We compare classification scores in Figure 6.1 for within language classification from the bolded diagonal of Table 6.7. For out of language classification, Figure 6.3 shows the average AUC-PR scores with the standard error of the mean. From our figures and from statistical significance tests showing no difference at the 0.05 level, but we conclude that data sampling has a slightly decreasing effect on the classification scores as seen in the results.

6.5.6 Comparing Different Feature Sets

We compare our proposed features and features directly from the PAN workshops by repeating our experiments with only these isolated sets of features. Our proposed diff based features are those described in Sections 6.3.1 and 6.3.3. We isolate these sets of features and repeated our experiments. We compare the combined classification scores (as in Table 6.7 for all features) for different subsets of features.

We present plots summarising the classification scores in Figure 6.2 for classification within the same language, and Figure 6.4 for the average scores for classification out of language. We include our experiments on the PAN baseline data sets as a comparison.

For within language classification, our proposed features have higher classification scores compared to previously used features across all five languages. Similarly, for out of language classification, we also find higher average classification scores. This suggests some regularity of vandalism within the same language and across languages.

6.6 Results of Related Work

We collect results of related work in Table 6.8, where AUC results are available. We compare these results within the context of knowing differences in data sets, sampling, and classifiers. We select the most appropriate results for comparison where possible, such as results for the Random Forest classifier, and similar sets of features. AUC-ROC gives the probability that a random sample revision containing vandalism will be ranked higher than a normal revision. This differs from AUC-PR, but both measures are related and AUC-PR is a more appropriate evaluation method for imbalanced classes [Davis and Goadrich, 2006].

The ratios of normal revisions to vandalised revisions are 10:1 and 13:1 for PAN 2011 and PAN 2010, respectively. Hence we observe high AUC-ROC scores because

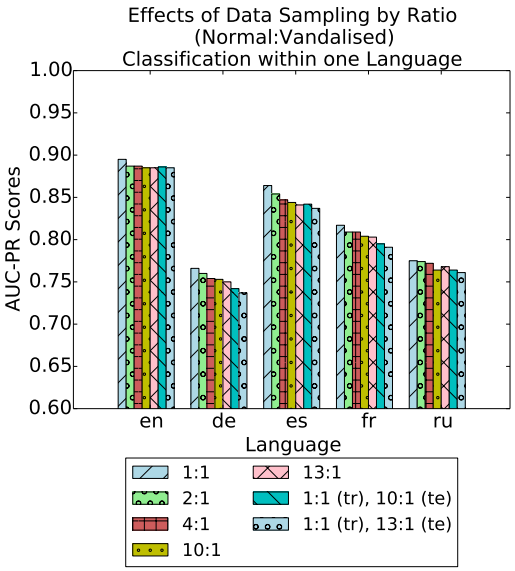

Figure 6.1: Comparison of different data sampling ratios for within one language classification. Values are from diagonals of Table 6.7 for ratio 1:1, (tr)ain and (te)st sets. Tables of results for other ratio tables not presented. Ratios 10:1 and 13:1 are similar to the ratios in the PAN 2010 and PAN 2011 data sets.

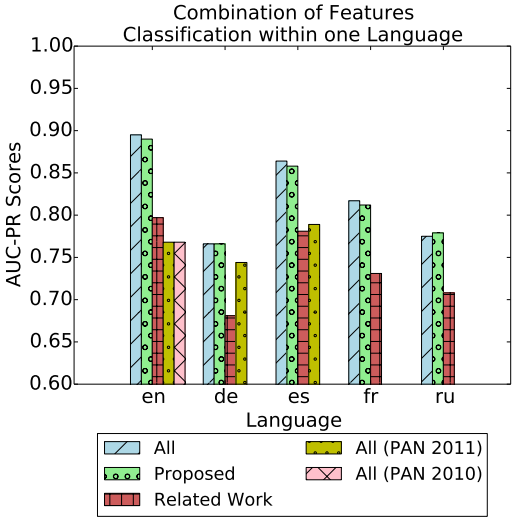

Figure 6.2: Comparison of different feature combinations for within one language classification. Values from diagonals of Table 6.3 (all features - PAN 2011 data set.), Table 6.7 (all features), and other tables for the proposed features (Sections 6.3.1 and 6.3.3) and related work features (Section 6.3.2), which are not presented.

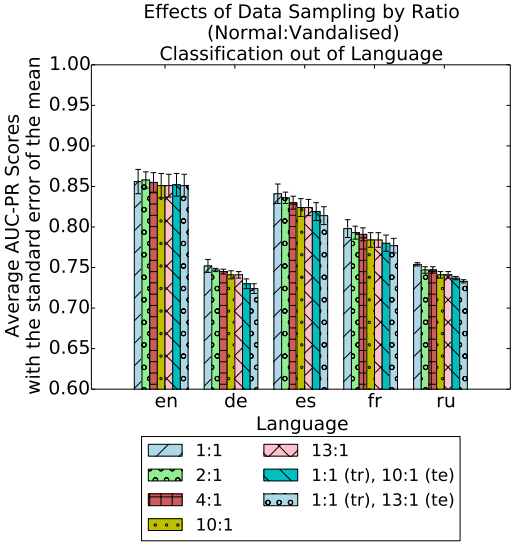

Figure 6.3: Similar to Figure 6.1, but for out of language classification results.

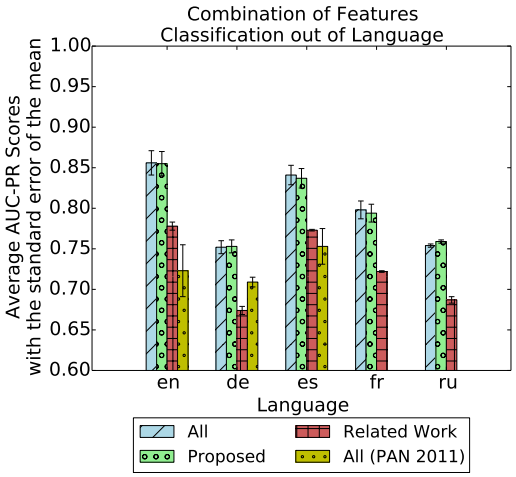

Figure 6.4: Similar to Figure 6.2, but for out of language classification results.

of many non-vandalism cases being correctly classified. Looking at the AUC-PR for the PAN data sets, our classification results in Table 6.7 are higher for the matching languages, suggesting a comparable performance in identifying true cases of vandalism, at the cost of a lower recall rate (seen in lower AUC-ROC scores).

The lower AUC-ROC scores for the classifiers suggest we may have more false positives of vandalism. For the comparable scores in Table 6.8 that use the same data sets, we believe the reasons for the lower AUC-ROC scores for our techniques may be explained by our focus on the cross-language aspect, where our features are at a disadvantage because they need to perform well across different domains of human languages. Furthermore, the imbalance nature of our Wikipedia data sets suggested a more appropriate priority of improving AUC-PR scores because these scores are more appropriate for evaluating classifiers on imbalanced data sets (see Section 2.7). Our results show that we have obtained classification performance comparable to related work (see Table 6.8), while demonstrating the differences between bots and users, and learning across languages.

West and Lee [2011] evaluated their set of vandalism features on the multilingual PAN-WVC-11 data set. The classifiers are evaluated within the same language, showing a lower AUC-PR score when applied on German and Spanish. This suggests the range features as developed with the English samples as the focus may be too broad, or simply are not suited to the differences seen in the German and Spanish samples. Our set of text-based features shows high AUC-PR scores for Spanish.

The research in this chapter continues from Chapter 5, where we developed Wikipedia vandalism data sets from article revisions and views. The data sets are balanced for classification and contain only metadata and temporal features. In Chapter 5, we did not consider the contribution of bots, nor looked at the content features for vandalism detection. We focused only on content features in this work mainly because we see these features as better discriminators of bots and users, because vandalism detection is mainly conducted on content only. In future work, we plan to incorporate metadata features to further analyse differences between bots and users.

6.7 Discussion

Vandalism is an increasingly important and urgent issue on all language editions of Wikipedia as Wikipedia's popularity and number of articles grows. Bots – used as force multipliers for maintenance tasks – have become essential to Wikipedia users in managing the influx of activity since 2006 [Halfaker and Riedl, 2012; Geiger, 2011]. The granting of editing capabilities to bots has allowed bots to become the power editors on Wikipedia [Adler et al., 2008]. As bots take the lead from users in identifying vandalism on the English Wikipedia, this maintaining of quality is deterring new and experienced editors [Halfaker et al., 2011]. Counter-vandalism bots may be solely responsible for the decline in the retention of new contributors because of their strict enforcement and poor communication of policy [Priedhorsky et al., 2007; Halfaker et al., 2011].

Table 6.8: Results of related work. Note that there are significant differences in data sets and techniques.

Source	Data set	AUC-PR	AUC-ROC
[Chin and Street, 2012]	Webis-WVC-07 (All)	0.643	0.663
[Adler et al., 2010, 2011]	PAN-WVC-10	0.737	0.958
[Mola-Velasco, 2010] [Adler et al., 2011]	PAN-WVC-10	0.731	0.946
[West and Lee, 2011] [Adler et al., 2011]	PAN-WVC-10	0.525	0.915
[Javanmardi et al., 2011]	PAN-WVC-10	-	0.955
[Harpalani et al., 2011]	PAN-WVC-10	-	0.930
[Adler et al., 2011]	PAN-WVC-10 (Text)	0.732	0.953
	PAN-WVC-10 (All)	0.853	0.976
Section 6.5.1	PAN-WVC-10 (en)	0.768	0.678
[West and Lee, 2011]	PAN-WVC-11 (en)	0.822	0.953
	PAN-WVC-11 (de)	0.706	0.969
	PAN-WVC-11 (es)	0.489	0.868
Table 6.3	PAN-WVC-11 (en)	0.768	0.684
	PAN-WVC-11 (de)	0.744	0.658
	PAN-WVC-11 (es)	0.789	0.716
Chapter 5	Train(en), Test(en)	0.902	0.872
	Train(de), Test(de)	0.871	0.795
Table 6.7	Train(en), Test(en)	0.895	0.858
	Train(de), Test(de)	0.766	0.688
	Train(es), Test(es)	0.864	0.818

While the media bolster approvals of counter-vandalism bots[4], signs of frustration by users are appearing in social media outlets such as Reddit[5] and Facebook[6]. This led us to investigate the differences between bots and users in the task of identifying vandalism with the overall aim to develop more accurate vandalism detection bots based on features and user identified cases of vandalism.

Our results show that distinguishing the vandalism identified by bots and users show statistically significant differences in recognising vandalism identified by users across languages, but there are no differences in recognising the vandalism identified by bots. This shows humans recognise a wider range of vandalism patterns beyond the capabilities of bots with our considered set of features, which suggests humans

[4]BBC News Magazine, "Meet the 'bots' that edit Wikipedia", 25 July 2012. http://www.bbc.co.uk/news/magazine-18892510

[5]Reddit user comments on a study of Wikipedia losing English-language editors, created on 4 January 2013. http://www.reddit.com/r/wikipedia/comments/15z5b8/wikipedia_losing_englishlanguage_editors_study/

[6]A Facebook page titled "Petition to get rid of Cluebot NG - Wikipedia", created on 25 December 2012. https://www.facebook.com/PetitionToGetRidOfCluebotNgWikipedia.

provide a deeper analysis of vandalism, resulting in identification of more difficult to detect types of vandalism. While this result is intuitive, we now have evidence of bots identifying similar vandalism to users. This suggests bots are becoming more sophisticated by handling more and more non-obvious cases of vandalism.

The benefits of cross-language learning of vandalism is to generalise classification models to Wikipedia languages without sufficient cases of identified vandalism to learn from. Our results show that learning from languages with many instances of vandalism, such as English, does generalise well to smaller Wikipedia languages. This means past and future work on feature engineering for vandalism detection in the English Wikipedia can be used on other languages without statistically significant loss in classification quality. Our results also show that related languages (such as English and German, and Spanish and French) are less affected by cross-language learning, where classification quality seems to be dependent on the target language.

An advantage of our approach is immediate text analysis of a revision with its previous revision to determine vandalism. We do not need additional metadata, derived data, and profiling of users to determine vandalism. Our new text-based features show comparable performance and improve on work that was based on samples of Wikipedia revisions. Our chosen features are specifically designed to generalise to the languages considered, which is reflected in the classification performance.

A limitation of our work is its reliance on text features, which may not capture vandalism that is apparent when looking at metadata and user reputation features. Our classification method uses an undersampling method to balance and reduce the size of the training data set. However, in Section 6.5.5 we have shown that undersampling does not statistically affect classification results in a significant way by repeating experiments with different training and testing ratios. We have shown the performance of only one classifier, which although is commonly used for vandalism research, may not be the best for cross-language learning as our results show in Chapter 5. Our sets of features are language independent only for the languages considered. For some languages, such as Mandarin Chinese, many word based features are no longer useful because of tokenisation issues and differences in the language. It is evident from the poor performance of the Russian language model that other techniques or features need to be developed that are suitable for the language. Vandalism is handled differently in each language community, and research is needed for non-English and especially non-European languages.

However, as we incorporate more and more human-like detection methods into bots, we believe that there may be diminishing returns. Bots function well on quantifiable patterns of vandalism, which are determined by humans as they capture commonalities of new types of vandalism into new features. The text data limits the types and creativeness of vandalism, which may be mostly quantifiable given sufficient historical data. There are very creative and subtle types of vandalism that will never be quantifiable, but as the complexities of these grow, we believe that they will be manageable by humans. Essentially, the development of this research is to automate as much detection of vandalism cases as possible to the point where the

remaining cases are feasible to be manually analysed by humans.

Overall, we have answered our research questions with some interesting results. Our evaluation over all revisions of each Wikipedia language shows more comprehensive and better results than sampling. We have shown bots and users differ in identifying vandalism, and that contributions of bots are important when analysing vandalism on Wikipedia. From our discussion, the trust of users in bots is lacking [Geiger, 2011], despite the high recognition of vandalism by bots. As we build better counter-vandalism bots, we will also aim to develop social aspects of bots to gain the trust of Wikipedia users [Halfaker and Riedl, 2012].

6.8 Summary

In this chapter, we have presented a comparison of bots and users in the vandalism detection task on Wikipedia across five languages. Vandalism is a major issue on Wikipedia, where bots are increasingly being used. We compared how bots and users differ in their identification of vandalism by learning from their identified cases. We developed text features that include features commonly used in vandalism detection tasks, and use the classifier to rank these features by their importance to bots and users across different languages. We generated training and testing data sets based on languages and editor type, and evaluated the classifier on their combinations. We showed and discussed differences in the identification of vandalism between bots and users across different languages. Our comparison to related work showed that our techniques are comparable and often achieve better performance on the entire Wikipedia data set compared to previous research. Our contributions showed we can learn vandalism from one Wikipedia language and apply a classifier to other languages with only a small loss in classification quality. Contributions of bots need to be acknowledged in research as bots are essential tools for Wikipedia to manage content quality.

In future work, we plan on looking at the contributions of anonymous users in identifying vandalism, as they are an understudied group of users because of difficulties in assigning an identity. The languages we chose are closely related to each other, so we would like to explore different languages, such as Arabic and Mandarin Chinese to complete the United Nations working set of languages. Non-European languages may need very specific techniques in tokenisation or specific features need to be developed for vandalism detection. Our ultimate aim is to build the human specific vandalism detection capabilities into the next generation vandalism detection bots based on machine learning approaches that can work effectively across many languages. An online system would also allow us to better gauge the effectiveness of these features and whether the performance drops are truly statistically significant.

The next two chapters build on the work of this chapter in two different paths of research. Chapter 7 investigates a novel vandalism detection technique for sneaky types of vandalism and compares to the feature-based detection technique shown in this chapter. The classification models of these novel techniques are also appli-

cable across languages. Chapter 8 shows the applicability of text features used for vandalism detection to a different domain of research on detecting malicious spam emails. The text features showing malicious modifications (vandalism) also allow patterns of text in spam emails to show the maliciousness of their file attachments and URLs. These two lines of research show the future of vandalism detection techniques and their extensibility to other domains of research on detecting malicious online activities.

Chapter 7

Context-Aware Detection of Sneaky Vandalism across Languages

In this chapter, we introduce a novel context-aware and cross-language vandalism detection technique for sneaky vandalism. We define sneaky vandalism as subtle types of vandalism that change the meaning or are out-of-context of the sentence being changed. Our detection technique is scalable to the size of the full Wikipedia, and extends the types of vandalism detectable beyond past feature engineering based detection techniques. Our technique uses word dependencies to identify vandal words in sentences by combining part-of-speech (POS) tagging with a conditional random fields (CRF) classifier.

This chapter is structured as follows. Section 7.1 introduces and motivates the need for context-aware detection techniques. Section 7.2 describes our additional data processing needed from Chapter 2 for this detection technique. Section 7.3 explains how POS tagging provides contextual information in edited sentences. Section 7.4 describes a CRF classifier and how it can be used to detect sneaky vandalism. Section 7.5 details our extensive experimental results – with the PAN and the full Wikipedia vandalism repairs data sets in five languages described in Chapter 2 – to evaluate our context-aware detection technique, and compare and contrast differences with a text feature based classification technique following from Chapter 6. Section 7.6 discusses our results, and advantages and limitations of our approach. Finally, Section 7.7 summarises and concludes this chapter and provides future directions for research.

7.1 Introduction

The introduction and prevalence of counter-vandalism bots since 2006 [Geiger, 2011] have reduced the exposure time of vandalism and the extra work needed by editors to repair vandalism [Kittur et al., 2007]. Vandalism detection research has introduced new techniques that improve the detection rate. These techniques often focus on developing features as input to machine learning algorithms [West et al., 2010a; West and Lee, 2011; Javanmardi et al., 2011]. A variety of features based on the metadata, editor characteristics, article structure, and content of Wikipedia articles have shown

to be effective in distinguishing normal revisions and revisions containing vandalism as we showed in Chapters 5 and 6. As new vandalism detection techniques are integrated into counter-vandalism bots on Wikipedia, vandalism of article content continues to become more sophisticated to avoid detection.

Wikipedia defines sneaky vandalism[1] as difficult to find, where the vandal may be using concealment techniques such as pretending to revert vandalism while introducing vandalism, or subtle changes in the article text that aim to deceive other editors to be legitimate changes. Subtle changes can be identified as vandalism because they may break consistency of descriptions across neighbouring articles and over time, deviate from common or correct grammatical structure, introduce uncommon word patterns, or change the meaning of a sentence. Text features used in vandalism research do not inherently capture the context of the sentences being edited as they do not consider word dependencies [Ramaswamy et al., 2013].

In this chapter, we propose a novel vandalism detection technique that is context-aware by considering word dependencies, and is scalable to the full Wikipedia. Our technique focuses on a particular type of sneaky vandalism, where vandals make sophisticated modifications of text that changes the meaning of the sentence without obvious markers of vandalism. We use a part-of-speech (POS) tagger [Schmid, 1994] to tag types of words in sentences changed in each edit, and conditional random fields (CRF) [Kudo, 2013; Lafferty et al., 2001] to model dependencies between tags to identify vandalised text.

We hypothesise that sneaky vandalism is out of context of sentences on Wikipedia, but seem normal with respect to the text features used in vandalism detection research. We evaluate our technique on the PAN data sets with over 62,000 edits, commonly used by related research; and the full vandalism repairs data sets with over 500 million edits of over 9 million articles from 5 languages: English, German, Spanish, French, Russian. These data sets are described in detailed in Section 2.5, where additional processing is needed in chapter. As a comparison, we implement a feature engineering classifier and analyse both classification results similar to Chapter 6, where and why our context-aware technique differs, and trade-offs of each type of classifier. Our results show how context-aware detection techniques can become a new state-of-the-art counter-vandalism tool for Wikipedia that complements current feature engineering based techniques.

Our contributions are (1) developing a novel context-aware vandalism detection technique; (2) demonstrating how our technique is scalable to the entire Wikipedia data set; (2) demonstrating the cross language application of classification models and the relationships between the languages considered; (3) replicating our experiments on the smaller PAN data sets often used in related work; and (4) demonstrating how our technique differs and contributes to traditional feature engineering approaches.

[1] http://en.wikipedia.org/wiki/Wikipedia:Vandalism

Table 7.1: Number of edits in different Wikipedia languages that have made changes to at least one sentence, split by type. "all" means combination or union of all data sets.

Data Set		Normal	Vandal Repairs
Wiki	en	256,796,879 (98.4%)	4,909,181 (1.9%)
	de	52,895,509 (99.7%)	164,097 (0.3%)
	es	31,742,769 (99.0%)	330,135 (1.0%)
	fr	41,657,071 (99.5%)	189,849 (0.5%)
	ru	24,335,713 (99.8%)	39,234 (0.2%)
	all	407,427,941 (98.6%)	5,632,496 (1.4%)
Data Set		Normal	Vandal Cases
PAN	2010 en	23,025 (92.7%)	1,804 (7.3%)
	2011 en	6,876 (89.1%)	844 (10.9%)
	2011 de	7,359 (95.1%)	381 (4.9%)
	2011 es	6,922 (89.7%)	792 (10.3%)
	2011 all	21,157 (91.3%)	2,017 (8.7%)

Table 7.2: Number of sentences extracted from edits.

Data Set		Normal	Vandal Repairs
Wiki	en	1,642,267,638 (96.6%)	58,183,825 (3.4%)
	de	370,010,973 (99.5%)	1,805,862 (0.5%)
	es	161,871,444 (98.9%)	1,879,431 (1.1%)
	fr	248,064,661 (99.3%)	1,671,695 (0.7%)
	ru	202,672,387 (99.6%)	747,854 (0.4%)
	all	2,624,887,103 (97.6%)	64,288,667 (2.4%)
Data Set		Normal	Vandal Cases
PAN	2010 en	236,721 (96.4%)	8,967 (3.6%)
	2011 en	82,256 (94.9%)	4,396 (5.1%)
	2011 de	80,308 (98.7%)	1,085 (1.3%)
	2011 es	42,998 (85.3%)	7,418 (14.7%)
	2011 all	205,562 (94.1%)	12,899 (5.9%)

7.2 Wikipedia Data Sets

In this section, we describe our construction of the PAN and Wikipedia vandalism repairs data sets for context-aware detection. We follow the parsing and vandalism identification techniques described in Section 2.5. We store each sentence and its word label n (normal) or v (vandal) for each diff along with some revision meta-data such as article name, revision number and time, and editor, to manually verify correctness and compare classification results. We discard vandal and normal edits that do not have a change in a sentence (e.g. edits with only additions or deletions). This different data processing resulted in fewer revisions, but our focus is on the numerous sentences changed during edits.

We do not consider changes such as mass additions or mass deletes because this technique is not designed for these types of vandalism. Consider a long article that has been blanked and replaced with 3 words, the order of POS tags in the sentence would play no role if the 3 words do not correspond to any paragraph. However, we have filtered out this case in our data processing. If these 3 words are part of a blanked paragraph, then they would satisfy the sentence edit criteria, but only if they align correctly as determined by the diff algorithm. The diff algorithm do check for matching sentences based on the common words. We have not specifically addressed edge cases where a vandal partially blanks a paragraph. These edge cases are better detected using feature-based techniques.

Table 7.1 shows the number of edits obtained from our data processing for the full Wikipedia and PAN data sets. Before processing, we have over 500 million revisions for all Wikipedia data sets and over 62,000 revisions for all PAN data sets, whereas after processing, we have approximately 400 million revisions for all Wikipedia data sets and approximately 47,000 revisions for the PAN data sets. This suggests approximately 20% of all revisions from both data sets do not contain edits that change a sentence, but edits that only make additions or deletions to the article. We find the percentage of vandalism repairs or cases are similar to those reported in Tables 2.3 and 2.4 from Chapter 2.

Table 7.2 shows the number of sentences extracted from the edits in Table 7.1. We see generally higher percentage of sentences in vandalism repairs or cases compared to normal sentences. This suggests often more than one sentence is vandalised in vandal edits that change sentences. Our context-aware detection technique learns vandalism and evaluates classification models from these sentences. We then remap these sentences to their edits to compare classification results with a text feature based detection technique.

To illustrate our data set, sneaky vandalism, and our detection technique, we present a running example below in Figure 7.1 which continues on in the next sections on POS tagging and CRF classifier.

7.3 Part-of-Speech Tagging

Before employing our context-aware classification technique, we process the sentences further and tag each word with descriptive information that allows our classifier to exploit contextual information. Our word tagging technique uses part-of-speech (POS) tagging, where the aim is to place words from a text corpus into text categories [Schmid, 1994]. The relationship of words in a sentence can determine their meaning and their language text categories. For example, "will" may be interpreted as a future tense verb, or a noun as in a first name or a legal document.

Martinez [2012] describes the state-of-the-art POS tagging methods: rule-based, where humans manually develop a set of language rules for rule-based POS taggers; transformation-based learning, where the rules are automatically constructed (or learned) from corpora, reducing the need for human experts in the previous

We present a fictitious example sentence[a] with sneaky vandalism to illustrate our tagging and classification technique in the following sections:

- Normal: Bread crust has been shown to **have more dietary fibers and** antioxidants.
- Vandalised (with word label): Bread (*n*) crust (*n*) has (n) been (*n*) shown (*n*) to (*n*) **make** (*v*) **hair** (*v*) **curlier** (*v*) **because** (*v*) **of** (*v*) antioxidants (*n*).

where the bolded words are changed words identified in the sentence diff that shows a vandalism repair (*v*) or normal edit (*n*) indicated by the revision comment, and the accumulated labels and tags for each word are contained in the parentheses.

[a] Adapted from http://en.wikipedia.org/wiki/Bread.

Figure 7.1: POS labelling example.

method; Markov model, where probabilistic (or statistical) models are constructed from the observed transitions of tag sequences from sentences – this technique has transformed the natural language processing domain; maximum entropy, where the history of observed tag sequences in a corpus is used to predict unseen tag sequences and to infer tags on unknown words with high accuracy; and a variety of other methods that include support vector machines, neural networks, decision trees, and finite state transducers.

We use the TreeTagger[2] software [Schmid, 1994] which uses binary decision trees to estimate the transition probabilities of POS tags and select the most appropriate tag from the available training data. Schmid [1994] and Schmid [1995] demonstrate that TreeTagger improves on Markov model based taggers by using decision trees to estimate transitional probabilities. By using decision trees, these probabilities become more reliable as the size of context for words can be adjusted by trimming the decision trees. TreeTagger also achieves high tagging accuracy and scales well with large corpora. The TreeTagger software provides many POS tag sets for different languages from many different contributors in the computational linguistics community. These tag sets include our working set of languages with complete parameter files and trained models.

For each sentence in our data sets, a POS tagger analyses known words (trained from a large manually labelled corpus) and assigns each word the most probable tag that describes it. In sneaky vandalism cases on Wikipedia, small changes can alter the meaning of sentences while not disrupting the correctness of text patterns in words (spelling) or sentences (grammar). For non-matches or unknown words, the TreeTagger software labels them as nouns. Handling non-matches or unknown words is a research area for POS tagging. It is a difficult task that requires inference from the surrounding tags or the morphological properties of the word. The TreeTagger

[2] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

software is trained on data sets and uses both methods.

Our example in Figure 7.1 illustrates this sneaky vandalism case, where in Figure 7.2, we show the output of the tagging by TreeTagger. We describe only the relevant tags to our example from the full English tag set documentation[2]: coordinating conjunction (CC), preposition or conjunction (IN), adjective (JJ), adjective - comparative (JJR), noun (NN), noun - plural (NNS), to (TO), verb - base form (VB), verb - past participle (VBN), verb - 3rd person (VBZ). We train the context-aware classifier on these tag sequences to predict the sequence of labels.

Continuing our running example from Figure 7.1, we have labels generated by TreeTagger as:

- Normal (tag, word label): Bread (NN, *n*) crust (NN, *n*) has (VBZ, *n*) been (VBN, *n*) shown (VBN, *n*) to (TO, *n*) **have** (VB, *n*) **more** (JJR, *n*) **dietary** (JJ, *n*) **fibers** (NNS, *n*) **and** (CC, *n*) antioxidants (NNS, *n*).
- Vandalised (tag, word label): Bread (NN, *n*) crust (NN, *n*) has (VBZ, *n*) been (VBN, *n*) shown (VBN, *n*) to (TO, *n*) **make** (VB, *v*) **hair** (NN, *v*) **curlier** (JJR, *v*) **because** (IN, *v*) **of** (IN, *v*) antioxidants (NNS, *n*).

where the parentheses contain the accumulated labels and tags for each word that are to be used in the CRF classifier.

Figure 7.2: TreeTagger tagging example.

7.4 Context-Aware Vandalism Detection

Context-aware detection techniques are needed because some types of vandalism cannot be easily detected with feature engineering approaches [Ramaswamy et al., 2013]. Our running example illustrates a case of potential vandalism that would require a human editor to repair. Our vandalism example has no clear markers of vandalism such as vulgarities, odd letter patterns in words, or radical changes to text [Adler et al., 2011]. Metadata features about an editor's past behaviour can show the likelihood that an editor is a vandal suspect [West et al., 2010a]. Text analysis is needed as metadata does not provide means to identify vandalism introduced into articles. Thus, we look to develop context-aware automated vandalism detection techniques to identify sneaky vandalism similar to our example.

We have chosen to develop our context-aware model using conditional random fields (CRF) [Lafferty et al., 2001], a probabilistic undirected graphical model for segmenting and labelling sequence data. This model differs from the research study by Wang and McKeown [2010] by not using *n*-grams to leverage semantic context. The limitation of *n*-grams is as *n* becomes large, the full *n*-gram model becomes large, but this is not the case with CRF as its conditional model allows overlapping features; this is further detailed in Sutton and McCallum [2010].

In this section, we briefly explain the theory, our application of CRF to vandalism detection, and advantages and disadvantages of CRF. The full development and derivation of CRF – with comparisons to hidden Markov models (HMMs) and maximum entropy Markov models (MEMMS) – are given by Lafferty et al. [2001], and additional models and discussion by Sutton and McCallum [2010].

From our processed data, we have for each sequence of words $\mathbf{s}$ (i.e. a sentence) and its word labels $\mathbf{l} = (l_1, l_2, ..., l_a)$ (i.e. n or v) and word tags $\mathbf{t} = (t_1, t_2, ..., t_b)$ (given by our POS tagger), where $a, b \in \mathbb{N}$. To exploit the contextual information of the sequence of word tags, we define three binary feature functions – on the training data sets – for three separate experiments:

$$f_j(l_k, \mathbf{t}), \quad g_j(l_{k-1}, l_k, l_{k+1}, \mathbf{t}), \quad h_j(l_{k-2}, l_{k-1}, l_k, l_{k+1}, l_{k+2}, \mathbf{t}) \tag{7.1}$$

The feature function f_j returns 1 when certain conditions – as learnt from the data set and explained below – are met, and 0 otherwise. This means for each tag, we define features that express some characteristics of the model only with its current label (f_j), or with the labels of the two adjacent tags (g_j) and the four (two on each side) adjacent tags (h_j).

For each feature function f_j, we assign weights θ_j that are also learnt from the training data sets through maximum likelihood estimation. Now, we can score a labelling $\mathbf{l}$ of tags $\mathbf{t}$ by summing the weighted features for each tag:

$$\text{sum}_k(\mathbf{l}|\mathbf{t}) = \sum_{j=1}^{m} \theta_j f_j(l_k, \mathbf{t}) \tag{7.2}$$

Note that feature function f_j can be interchanged with g_j or h_j, with differing function parameters. Then we transform the scores into probabilities similar to the joint distribution of HMMs [Sutton and McCallum, 2010]:

$$p(\mathbf{l}, \mathbf{t}) = \frac{1}{Z} \prod_{k=1}^{K} exp\left\{\text{sum}_k(\mathbf{l}, \mathbf{t})\right\} \tag{7.3}$$

where Z is a normalisation constant to keep $p(\mathbf{l}, \mathbf{t})$ between 0 and 1, which is cancelled in the fraction of the next step below.

Finally, we have the conditional probability that models the conditional distribution as a linear-chain CRF [Sutton and McCallum, 2010]:

$$p(\mathbf{l}|\mathbf{t}) = \frac{p(\mathbf{l}, \mathbf{t})}{\sum_{\mathbf{l}} p(\mathbf{l}, \mathbf{t})} \tag{7.4}$$

The training phase above gives us a model for each Wikipedia data set. To predict the labels (n or v) of a new input set of tags (e.g. POS) extracted from an unseen sentence, we compute:

$$\mathbf{l}^* = \text{argmax}_{\mathbf{l}}\ p(\mathbf{l}|\mathbf{t}) \tag{7.5}$$

which gives us the predicted tags (e.g. POS), which are combined with the true labels,

POS tags, and words of the sentence. Our running example below in Figure 7.3 illustrates possible predicted labels.

An advantage to using CRF in our application is the immediate identification of words predicted as vandalised. This is useful for further verification of changes in tags before and after edits by editors. Other advantages include specification of different tag sets, specifying dependencies across multiple tag sets, and developing additional more interesting labels such as "maybe vandal". In contrast with feature engineering approaches, overlaps or dependencies between features are either ignored, or feature selection or combination techniques are used. The recoverability of evidence for vandalism requires additional processing and inspection because of the global nature of feature engineering approaches.

A disadvantage of CRF is the potential slow convergence of training models when the feature functions are complex (long chains of adjacent tags or tags from multiple tag sets) or when they have strong dependencies. Other disadvantages include the difficulty in developing expressive tag sets suitable for the domain of the data (e.g. word semantics[3], WordNet[4], and Wikipedia ontologies[5]), limitation to discrete tags and labels, and difficulties in tuning with respect to the manual specification of feature functions or heuristics for automatically generated feature functions.

We use an implementation of CRF by Kudo [2013], named CRF++[6], to evaluate our vandalism detection technique. CRF++ is an open source implementation of CRFs that supports custom feature sets and labelling of sequential data amongst many other functionalities and applications.

In the training and testing data sets for CRF++, each word can have multiple tags from different tag sets, where any of these tag sets can be learnt and predicted. The encoding of contextual relationships between the word tags is manually specified through feature templates that allow CRF++ to generate feature functions (Equation 7.1). For each tag, these templates determine which other tags relative to the current tags are to be learnt from (Equation 7.2). In the training phase (Equation 7.4), a model of tag distributions is learnt, which can then predict the desired tags (Equation 7.5). These predicted tags can be evaluated for correctness and feature templates and tag sets can be adjusted as needed to improve results.

We process our data further as required by CRF++ and recover classification results of test sentences for each edit for further evaluation. Our resulting testing data sets resemble our example below in Figure 7.3, where we can now evaluate classification performance.

For the results, we stored a unique set of sentences to be evaluated, where each edit has a mapping to the evaluated sentences. We did a strictly bias-for-vandalism aggregation by labelling an edit to be vandalism if it contain at least one vandal labelled word. We do have results at the word level and sentence level, but it is a different task to identify vandalised words and sentences, compared to a whole

[3] https://www.freebase.com/
[4] http://wordnet.princeton.edu/
[5] http://dbpedia.org/About
[6] https://crfpp.googlecode.com/svn/trunk/doc/index.html

edit. In future work that delve further into this technique, we intend to compare different types of aggregation, such more lenient vandalism labelling where more than one vandal labelled word is needed or putting weights on the different words to determine their significance.

This final example continues from our example in Figure 7.2. Assuming we have trained the CRF on sentences from the Wikipedia data sets to obtain feature functions for the current tag and tags adjacent, then for classification, we may have an optimal labelling of our vandalised sentence as:

- Vandalised (tag, word label, `predicted label`): Bread (NN, *n*, n) crust (NN, *n*, n) has (VBZ, *n*, n) been (VBN, *n*, n) shown (VBN, *n*, n) to (TO, *n*, n) **make** (VB, *v*, v) **hair** (NN, *v*, v) **curlier** (JJR, *v*, v) **because** (IN, *v*, n) **of** (IN, *v*, n) antioxidants (NNS, *n*, n).

where the predicted labels are n and v, and the correct labelled vandal words are in bold text and coloured, where green means a correct label and red means an incorrect label of a vandal word.

The implications of these mislabellings are that they may be common phrases that are normal (as shown above), or incorrect patterns that need to be manually readjusted. We see the advantage of this context-aware detection technique by the immediate presentation and labelling of evidence, which could be displayed to a user interface for a user to inspect and validate, or readjust tag sets, feature functions, or detection parameters.

Figure 7.3: CRF classification example.

7.5 Results

We split each data set by the number of edits for 10-fold cross-validation. We performed the sampling before splitting the data set into folds. Each fold has small variations in the number of sentences, because sentences are not evenly distributed across edits. The same testing data are used for all data set samples as the sampling is conducted only on the normal edits. Due to time and resource constraints, we were unable to perform classification for the full Wikipedia vandalism repairs data sets, so we performed sampling with different ratios of normal edits to vandal repair edits. For example, "2-to-1" means 2 normal edits for every 1 vandal repairs edit. This also allows us to compare the effects of data sampling for context-aware classification techniques.

We present our classification results as the area under the precision-recall (PR) curve (AUC-PR) and as the area under the receiver-operator characteristic (ROC) curve (AUC-ROC). These measures are described in Section 2.7. Our plots compactly present our classification results, which show trade-offs in performance in each data set for each classifier.

7.5.1 CRF with Tree Tags

The CRF classifier in our first set of results is trained and tested on the same source and target language, or named as "within" language classification. CRF classification results for the PAN data sets are presented in Figure 7.4 and for the Wikipedia vandalism repairs data sets in Figure 7.5. We plot AUC-PR results against AUC-ROC results for different ratios of data samples, where each marker represents a different data set. For both data sets, we observe an expected decrease in AUC-PR scores as the sampling ratio increases because of the increase in false positives (FP) and false negatives (FN); and an increase in AUC-ROC scores where the false positive rate (FPR) is normalised with the true negatives (TN). For the PAN data sets, the results of sampling ratios converge on the results of using the full data set. The convergence of the Wikipedia sampled data sets suggests similar results for training on all data.

The CRF classification results for the PAN data sets in Figure 7.4 generally show consistent AUC-ROC scores for all data sets. The 2010 English data set (2010-en) shows consistently high results for both AUC-PR and AUC-ROC scores, which can be attributed to the relatively higher number of training data available compared to the 2011 data sets. The German data set (2011-de) has the lowest number of vandalism cases, where CRF is unable to model vandalism cases as seen with AUC-ROC results below 0.5 (worst than random guess). These poor results can be explained by our default labelling of unknown sentences as "vandalism", which in hindsight may have been too strong and affected the results negatively. In future work, we look to better label these unknown cases and create higher quality data sets similar to the PAN data sets, but for context-aware techniques. Combining all 2011 data sets ("all") shows an approximate average of the results for each 2011 data set. The different feature functions show minor improvements when using more adjacent tags for context.

The results for the Wikipedia data sets in Figure 7.5 show significantly higher AUC-PR and AUC-ROC scores than the PAN data sets for each ratio of sampled data sets. Non-English Wikipedias have much higher scores than the English Wikipedia, suggesting vandalism in non-English Wikipedias more often break sentence structure detectable through changes in the sequence of POS tags. In particular, POS tags seem to be most suited to the Spanish (es) Wikipedia because of the relatively higher AUC-ROC scores. The different feature functions show minor improvements to AUC-PR and AUC-ROC classification scores, similar to the PAN data sets. Combining all data sets ("all") shows scores highly similar to the English (en) results because of the overwhelming number of English vandalism cases as seen in Table 7.1.

We have included a lot of information on these graphs with the aim to summarise all of our results. There are differences between the graphs, which we tried to emphasise by the non-standard axes. The graphs for the Wikipedia data sets do show an improvement to using context, which is less clear than the non-improvement results of the PAN data sets. We tried to add axes lines to the plots, but they looked much more cluttered.

The difference between our work and Wang and McKeown [2010] is that we allow the CRF classifier to derive its own model of the vandalised sentences with

minimal guidance from a human user. In Wang and McKeown [2010], the feature-based techniques are inflexibility and require a user to guess or derive vandalism features from past results to detect new types of vandalism. Our exploratory work looks at new techniques that allow alternative classifiers to infer vandalism with minimal guidance from human users.

Overall, the relatively numerous vandalism examples in the full Wikipedia data sets allow the CRF classifier to better distinguish vandalism. The classification scores show relatively minor improvements when considering context by learning adjacent tags. We may not be observing higher improvements because our data selection constraints are not sufficiently strict to identify only sneaky vandalism. The proportion of sneaky vandalism to non-sneaky vandalism in our data sets is likely to be small because of the difficulty in detecting this type of vandalism. The CRF classifier is also highly dependent on feature functions [Sutton and McCallum, 2010], so more complex feature functions may allow higher improvements to classification scores.

7.5.2 Reusing Models Across Languages

We investigate the cross-language performance of our context-aware technique, where Wikipedia vandalism detection models are trained on one language and reused to classifying on other languages. Our POS taggers share some tags across our working set of languages, and the definition of CRF does not include a model for the probability of tags $p(\mathbf{t})$[7], which makes CRF suitable for classifying unseen tags [Sutton and McCallum, 2010]. In this section, we have not taken into account the differences in syntactic structures of different languages. In this exploratory work, we investigate how context-aware techniques handle unseen forms of vandalism, and leave improvements that take into account syntactic structures for future work.

For a target language, we reuse the CRF models trained in other languages and report the average classification results with one standard deviation. For example, for the English (en) target language, we reuse the German (de), Spanish (es), French (fr), and Russian (ru) models, and report the average and one standard deviation of all classification scores from 10-fold cross-validation. Our CRF classification results are shown in Figure 7.6 for the PAN data sets, and in Figure 7.7 for the Wikipedia data sets.

The PAN data sets show weaker classification scores compared to classification within the same language. The range of scores varies widely, especially for the AUC-ROC scores. The effect of using adjacent tags for context-aware classification is now seen through the reduction of the standard deviation. German (de) vandalism cases do not benefit from classification models from other languages as there are too few samples. Reusing CRF models trained on small data sets do not provide any significant benefits as observed by a lower convergence of average scores and clusters of results for the sampling ratios.

The Wikipedia data sets show stronger classification scores compared to the PAN

[7] From the joint distribution of HMMs, which is often difficult to model because $p(\mathbf{t})$ may contain highly dependent features [Sutton and McCallum, 2010].

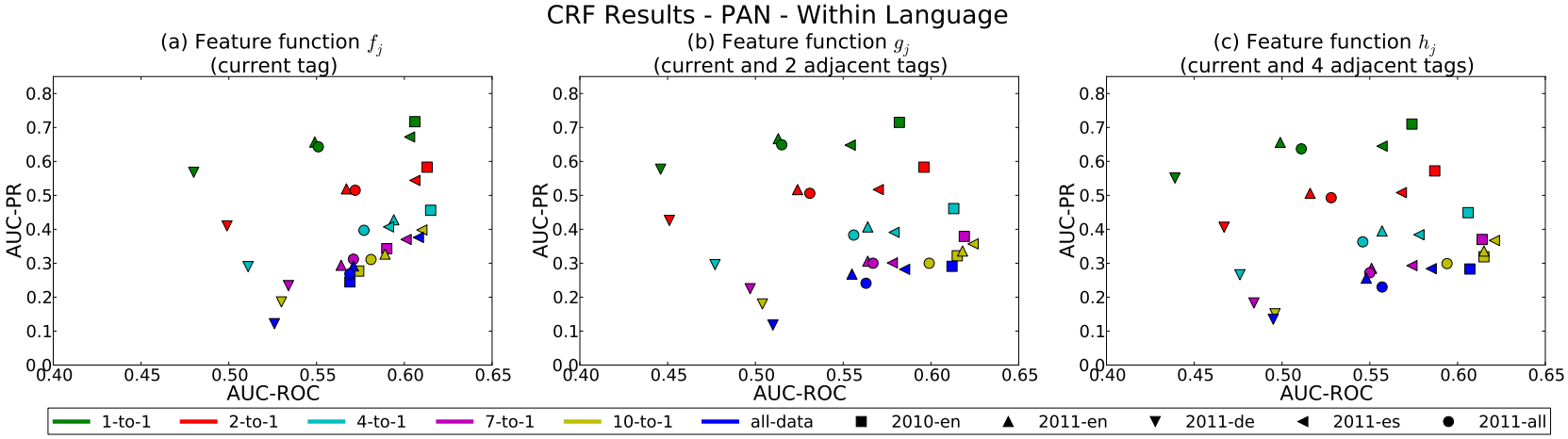

Figure 7.4: CRF results for classification within the same language on the PAN data sets. Upper right is better.

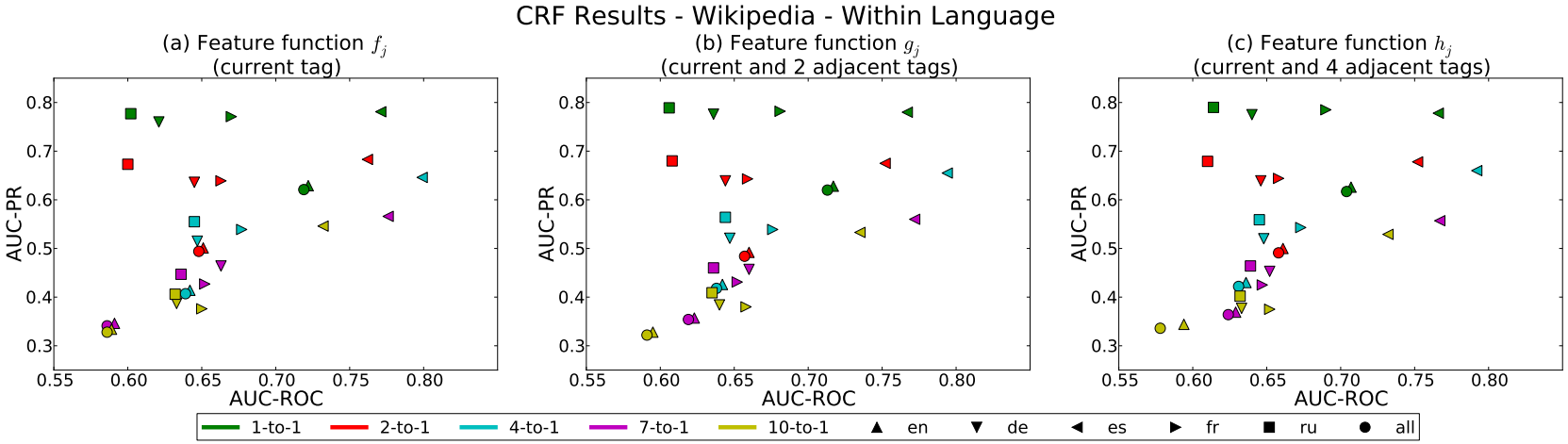

Figure 7.5: CRF results for classification within the same language on the Wikipedia vandalism repairs data sets. Upper right is better.

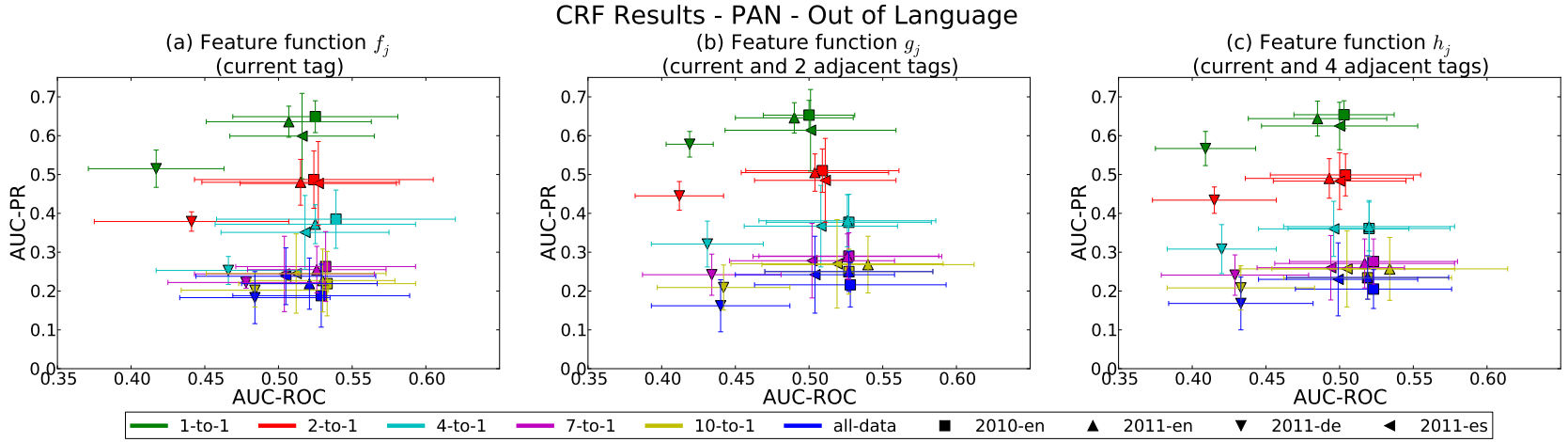

Figure 7.6: CRF results with one standard deviation for out of language classification on the PAN data sets. Upper right is better.

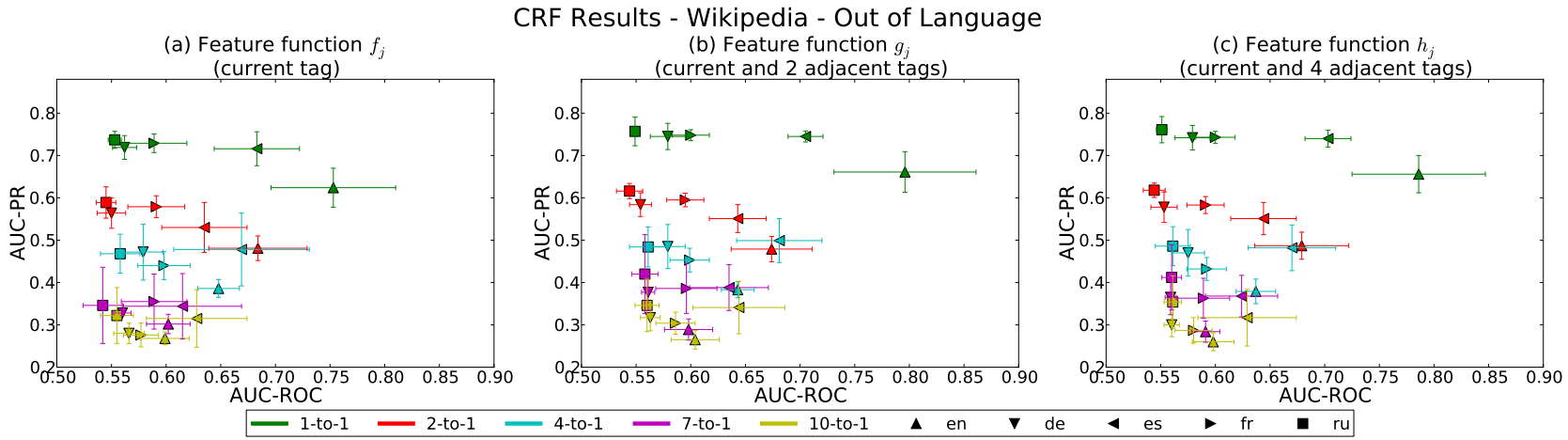

Figure 7.7: CRF results with one standard deviation for out of language classification on the Wikipedia vandalism repairs data sets. Upper right is better.

data sets, similar to within language classification. The feature functions with more adjacent tags also reduce the variance in the standard deviation, especially for AUC-PR scores. This suggests the CRF classifier is more precise in classifying vandalism cases when it has contextual awareness of other tags. An interesting result is the higher classification scores for the English (en) target language (especially for the 1-to-1 sampling ratio), where scores of within language classification do not fall into one standard deviation across many data ratios. The non-English CRF models may be identifying sneaky vandalism that is lost within the English CRF model because of the large size difference in the training data sets.

Overall, CRF models can be reused across languages – provided there is a sufficient amount of training data – with a small loss in classification performance, but also some highly beneficial gains. By using adjacent tags, the CRF classifier has less variance in both AUC-PR and AUC-ROC scores, suggesting CRF models are more precise. The small sizes of the PAN data sets severely limit the quality of the models to be reused across languages. With the large Wikipedia data sets, we can reuse models across languages with low variance of AUC-PR and AUC-ROC scores, where we show beneficial gains for detecting vandalism in the English Wikipedia.

7.5.3 Comparing to Feature Classification

As a comparison to our context-aware technique, we implement a feature engineering based classifier following Chapter 6 and related work [Adler et al., 2011; Javanmardi et al., 2011; Mola-Velasco, 2010; West and Lee, 2011]. Similar related work such as González-Brenes et al. [2011] has also compared the differences between using feature-based representations compared to sequence representations (i.e. CRF) for classification. One of the limitations of feature-based techniques is the inflexibility of features and the feature engineering required to capture the variance of text [González-Brenes et al., 2011]. We could propose and derive many types of features for vandalism detection, but this is a continually more difficult task as the nature of vandalism changes and the features become more complex. Thus, we look to exploit the adaptability of context-aware techniques in detecting vandalism.

Table 7.3 presents features investigated in our previous work in Chapter 6. We select a relevant subset of features from winning entries of the PAN workshop competitions (features `P01-PW` to `P12-LZW`), and contribute our own subset of features (features `F01-NWD` to `F12-WS`). We calculate these features from the same vandalism repairs data sets presented in Section 7.2, and follow the same 10-fold cross-validation technique.

The PAN workshop features analyse the words changed in an edit for abnormal variations in text that might indicate vandalism. Features `P01-PW` to `P03-SW` show three types of words that are common or indicative of vandalism: pronouns, slangs, and vulgarities. These features are described in detail in Section 6.3.2, where we use the same methodology. Features `P04-CW` to `P11-SC` count the different word types. By looking at the letters of each word, some indications of possible vandalism are uppercase words, words with digits, and words that are single letters. These features

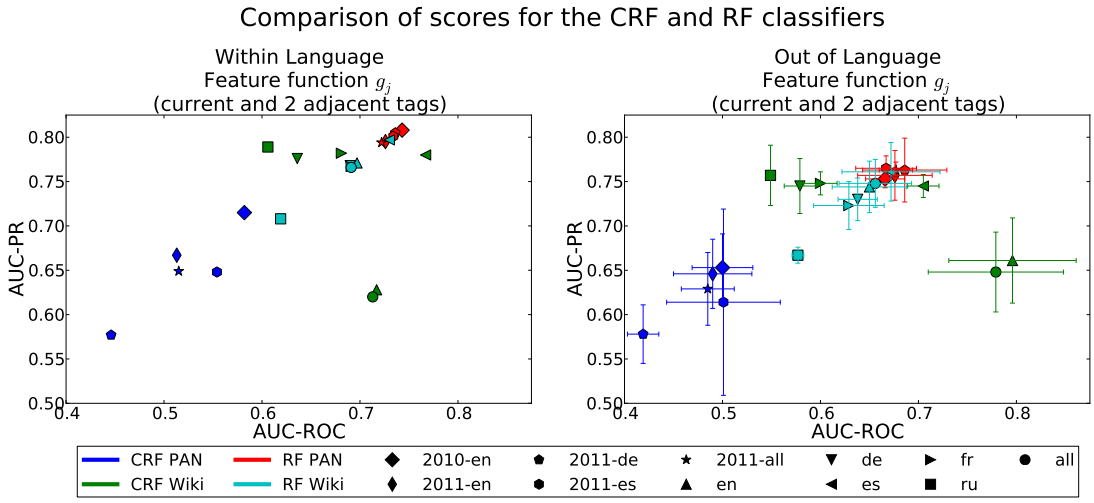

Figure 7.8: Comparison of scores for the CRF and RF classifiers.

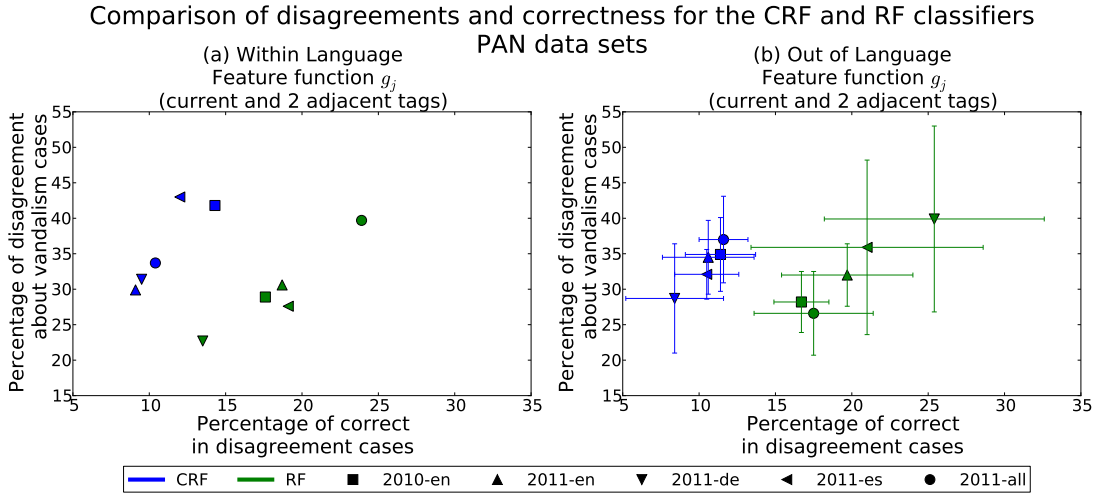

Figure 7.9: Comparison of classifier disagreements and correctness for the PAN data sets.

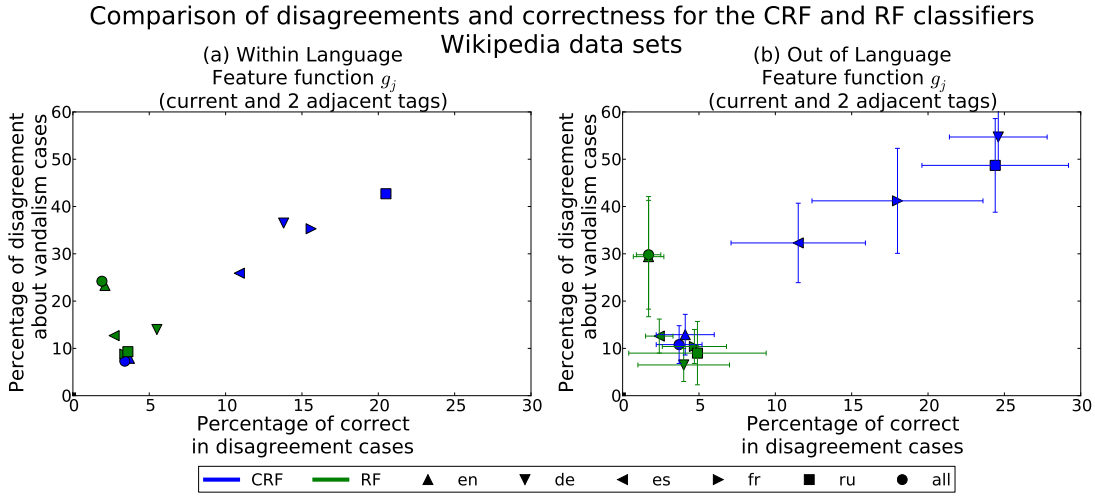

Figure 7.10: Comparison of classifier disagreements and correctness for the Wikipedia vandalism repairs data sets.

are common indicators of vandalism in related work [Mola-Velasco, 2010; West and Lee, 2011].

Our contributed features analyse additional variations of changes to words in an edit. Features `F01-NWD` and `F02-TWD` count the words changed in an edit. Features `F03-UL` to `F07-CD` look at the ratios of letters to words. We select these features with definitions from Mola-Velasco [2010], but apply them with modifications to the equations as needed to suit the word level instead of the document level. We take the maximum or minimum of these ratios for each word as a strong indicator of vandalism. Feature `F08-LRC` shows the length of the longest repeated character in a word as used in Mola-Velasco [2010], which is often a clear case of vandalism. To complement this feature, the compressibility of words can identify abnormally long repeated sequence of letters. We compare three compression algorithms and take the lowest compression ratio, indicating the highest compressibility of a word. Features `F09-ZLIB` and `F10-BZ2` are provided to extend and contrast the compression feature `P12-LZW` from Mola-Velasco [2010]. Features `F11-WL` and `F12-WS` count the longest unique words and the total size of the unique words in the sentence difference. These are intuitive features from Mola-Velasco [2010] and West and Lee [2011], but with a different interpretation and application.

We use the Random Forest (RF) classifier from the Python based Scikit-learn toolkit [Pedregosa et al., 2011]. This classifier was shown to be the most robust and generally best performing classifier from related works and in Chapters 5 and 6, hence we did not compare alternative classifiers. We present our comparison plots for the 1-to-1 data sampling ratio in Figure 7.8 for within language classification and for out of language classification.

For within language classification, the RF classifier has strong classification results for both PAN and Wikipedia data sets. For the PAN data sets, the RF classifier performs consistently well with features in Table 7.3, as expected from related work [Adler et al., 2011; Javanmardi et al., 2011; Mola-Velasco, 2010; West and Lee, 2011]. The tight cluster of RF PAN results (Figure 7.8) suggests the features are language independent and have strong performance. The RF classifier on the full Wikipedia data sets shows similar strong classification performance. The poor performance of the Russian (ru) data set may be due to the relatively fewer vandalism cases compared to other languages, or these features are not ideal for the Russian data set. The CRF PAN data set results are worst compared to all other results, showing the unsuitability of context-aware techniques on small data sets. The CRF and RF Wikipedia results show trade-offs in AUC-PR and AUC-ROC scores.

For out of language classification, we see a tight cluster of RF results for both the PAN and Wikipedia data sets (Figure 7.8). This is expected as within language classification shows highly similar classification scores. We explored the out-of-language classification using RF in Chapter 5 for metadata features and in Chapter 6 for text features, where both show a small loss in classification scores that is also observed here. Interestingly, the CRF and RF Wikipedia scores for the English (en) and "all" data set have almost opposite AUC-PR and AUC-ROC scores. This shows a trade-off in precision (P) and FPR when using each classifier. The CRF classifier has higher

Table 7.3: Features for feature engineering vandalism detection. Features `F01` to `F12` are our contributions.

Feature	Description
`P01-PW`	Pronoun words
`P02-VW`	Vulgar words
`P03-SW`	Slang words
`P04-CW`	Capitalised words
`P05-UW`	Uppercase words
`P06-DW`	Digit words
`P07-ABW`	Alphabetic words
`P08-ANW`	Alphanumeric words
`P09-SL`	Single letters
`P10-SD`	Single digits
`P11-SC`	Single characters
`P12-LZW`	Lowest compression ratio, lzw compressor
`F01-NWD`	Number of unique words
`F02-TWD`	Number of all words
`F03-UL`	Highest ratio of upper to lower case letters
`F04-UA`	Highest ratio of upper case to all letters
`F05-DA`	Highest ratio of digit to all letters
`F06-NAN`	Highest ratios of non-alphanumeric letters to all letters
`F07-CD`	Lowest character diversity
`F08-LRC`	Length of longest repeated character
`F09-ZLIB`	Lowest compression ratio, zlib compressor
`F10-BZ2`	Lowest compression ratio, bz2 compressor
`F11-WL`	Longest unique word
`F12-WS`	Sum of unique word lengths

TPR and FPR scores instead of the higher precision (P) scores of the RF classifier.

Overall there are differences between the classification results of the CRF and RF classifiers and also trade-offs between correctness and preciseness for each classifier. This comparison of different types of classifiers shows a new direction of vandalism detection research from the commonly seen feature construction based research.

7.5.4 Disagreements of Classifiers

We further investigate the differences between classifiers by analysing their disagreements in predicting vandal edits, and which classifier is correct in those disagreements. For each edit in the PAN and Wikipedia data sets, we recorded the true label (normal or vandalism) and the predicted labels from the CRF and RF classifiers. For the 1-to-1 sampling ratio, we calculate the percentage (of all edits) of a classifier disagreeing with the other classifier about whether an edit is a vandal edit and the percentage (of all edits) of cases where that classifier is correct.

We present the within and out of language disagreements of the CRF and RF classifiers for the PAN data sets in Figure 7.9 and the Wikipedia data sets in Figure 7.10. For the PAN data sets, the CRF classifier disagree more often against RF for an edit containing vandalism, but with a lower percentage of correctness. This suggests the CRF classifier is interpreting more edits as vandalism, resulting in more false positives (FP). The results are similar for out of language classification with large standard deviations. For the Wikipedia data sets, the CRF classifier is disagreeing more against RF, but it also has a higher percentage of correctness. The results are similar for out of language classification with large standard deviations similar to the PAN data sets.

Overall, the disagreement results are consistent with the comparison of CRF and RF classifiers in the previous section. The trade-offs in AUC-PR and AUC-ROC scores for each classifier are reflected in their disagreements, showing higher true positives (TP) or false positive (FP) cases of vandalism that influence the AUC scores. The false negatives (FN) are seen for the other classifier when the current classifier is correct. For example, if CRF predicts an edit is vandal and is correct, which disagrees with RF, then RF has a false negative (FN) result. The rate of false negatives (FN) is higher for the RF classifier in the Wikipedia data sets. The CRF and RF classifiers are identifying different types of vandalism, where they are complementing each other's detection technique.

7.5.5 Results of Related Work

We collect results of related work in Table 7.4 as in Chapter 6, where AUC scores are available. We compare these results within the context of knowing differences in data sets, sampling techniques, and classifiers. We select the most appropriate results relating to our presented results where possible, such as results for a RF classifier, and similar sets of features. We could not include results for Wu et al. [2010] and Ramaswamy et al. [2013] as they only presented separately in plots the F1, precision, and recall scores. Other related work do not consider word dependencies for detecting vandalism nor aim to develop detection techniques for sneaky vandalism.

The CRF classifier has seemingly poorer performance on both PAN and Wikipedia data sets compared to related works that employ numerous and some complex features in their classification techniques. This poorer performance is likely because of different types of vandalism being detected (as shown in Section 7.5.4 above), or a refinement of this (exploratory) technique is needed in future research. The advantages of using this context-aware vandalism detection technique are that they provide immediate and highlighted evidence of vandalism to users, where the users can validate or readjust the tag sets, feature functions, or other parameters of the CRF classifier. The different features and techniques are too significant with different trade-offs in classification performance. Our aim is to provide an overview of related works that have provided different vandalism detection techniques, without naively comparing raw results.

We have shown that a CRF classifier is a novel context-aware vandalism detection

Table 7.4: Results of related work. Note that there are significant differences in data sets, features, and techniques used.

Source	Data set	AUC-PR	AUC-ROC
Chin and Street [2012]	Webis-WVC-07 (All)	0.643	0.663
Adler et al. [2010, 2011]	PAN-WVC-10	0.737	0.958
Mola-Velasco [2010]; Adler et al. [2011]	PAN-WVC-10	0.731	0.946
West and Lee [2011]; Adler et al. [2011]	PAN-WVC-10	0.525	0.915
Javanmardi et al. [2011]	PAN-WVC-10	-	0.955
Harpalani et al. [2011]	PAN-WVC-10	-	0.930
Adler et al. [2011]	PAN-WVC-10 (Text)	0.732	0.953
	PAN-WVC-10 (All)	0.853	0.976
West and Lee [2011]	PAN-WVC-11 (en)	0.822	0.953
	PAN-WVC-11 (de)	0.706	0.969
	PAN-WVC-11 (es)	0.489	0.868
Chapter 5	Wikipedia (en)	0.902	0.872
	Wikipedia (de)	0.871	0.795
Chapter 6	Wikipedia (en)	0.895	0.858
	Wikipedia (de)	0.766	0.688
	Wikipedia (es)	0.864	0.818
Fig. 7.4(c) CRF 1-to-1	PAN-WVC-10 (en)	0.710	0.574
	PAN-WVC-11 (en)	0.656	0.499
	PAN-WVC-11 (de)	0.551	0.439
	PAN-WVC-11 (es)	0.645	0.557
Fig. 7.5(c) CRF 1-to-1	Wikipedia (en)	0.626	0.707
	Wikipedia (de)	0.775	0.640
	Wikipedia (es)	0.778	0.766

technique that provides additional benefits to feature engineering based classifiers. Our CRF technique is an exploratory technique demonstrating advantages and feasibility on the full Wikipedia data sets.

7.6 Discussion

Our context-aware vandalism technique for sneaky vandalism shows the feasibility of using part-of-speech (POS) tags and a linear-chain conditional random fields (CRF) classifier for vandalism detection. Our results show that the small PAN data sets may not contain sufficiently many cases of sneaky vandalism, which led to poor CRF results. In contrast, our extracted Wikipedia vandalism repairs data sets contain numerous sneaky vandalism cases, which have strong classification performance for within and out of language classification. By using more information from adjacent words in sentences for contextual awareness, small improvements in classification scores are seen. The CRF models can be reused across languages similar to our

demonstration in Chapters 5 and 6 for feature engineering based models, and repeated in our experiments for text features in Section 7.5.3. The reuse of CRF models shows that non-English models can improve the detection on vandalism on the English Wikipedia. Although the CRF classification scores are not as strong as the score obtained with the Random Forest (RF) classifier in most PAN and Wikipedia data sets, we show through analysis of disagreements between the classifiers that each is detecting different types of vandalism based on their correctness in disagreements. We show context-aware vandalism detection is most effective on large data sets, where they complement the types of vandalism detectable by feature engineering based approaches.

Our findings show a novel context-aware vandalism detection technique that scales to the size of Wikipedia. Past context-aware detection techniques inefficiently generated numerous word pairs to determine context in sentences [Wu et al., 2010; Ramaswamy et al., 2013]. Our technique allows descriptive word tags and complex relationships between these tags to be modelled to find increasingly sneakier types of vandalism. For example, more descriptive POS tags or cross language tagging may provide additional contextual information, and more complex dependencies between tags could be modelled through the feature functions for the CRF classifier. The CRF classifier allows the immediate identification of vandal words along with evidence, in contrast to the global nature of classification in feature engineering techniques. Our results show that context-aware vandalism technique may be better suited to large Wikipedia data sets, which may be a reason for lack of context-aware detection research on the small PAN data sets.

Some limitations of our proposed technique are from word tagging and disadvantages of using CRF. The tagging of words is complex and requires manually labelled corpora for optimal performance in different tagging techniques [Martinez, 2012]. The limited type of word tags (for TreeTagger) used in this chapter may not provide sufficiently descriptive contextual information for the CRF classifier. We used POS tags because they provide contextual information and have well supported tagging software, but other tags need to be investigated, such as word semantics, WordNet, and Wikipedia ontologies. Other limitations are from feature functions that highly influence CRF classification performance [Sutton and McCallum, 2010]. Feature functions that are complex such as long chains of adjacent tags, from multiple tag sets, or have strong dependencies can lead to slow convergence of CRF training models. The tuning and complexity of feature functions, or heuristics for automatically generated feature functions, have not been explored rigorously in this chapter. We hope to address these limitations in future work and further explore novel ways of detecting vandalism on Wikipedia.

7.7 Summary

In this chapter, we have shown a proposed novel context-aware detection technique for sneaky vandalism on Wikipedia based on a conditional random fields (CRF) clas-

sifier. We evaluated this classifier on two data sets, the PAN data sets commonly used by related works, and our own vandalism repairs data set built from all Wikipedia edits from five languages. We used part-of-speech (POS) tagging to tag all sentences changed in edits from both data sets, then used a conditional random fields (CRF) classifier to train and test on our data sets using 10-fold cross-validation. Our context-aware technique showed results comparable to related work using feature engineering based approaches. As a comparison, we developed a set of text features and detected vandalism using a random forest classifier on the same data sets, and analysed where and why both classification techniques disagree on certain edits. We have shown through our results that context-aware techniques can become a new counter-vandalism tool for Wikipedia that complements current feature engineering based approaches.

In future work, we aim to develop a language independent tag set that uses information from feature engineering approaches. Our working set of languages contains some shared POS tags, where we can unify these tags into higher level word tags that have direct mappings across languages, such as nouns, pronouns, verbs, adverbs, and adjectives. In a similar path of research, we would like to complete our analysis of the sequences of tags to better understand patterns that are indicative of vandalism. We plan to extend our linear-chain CRF to a general CRF that allows modelling of dependencies between articles, where vandals may also target adjacent internally linked articles. Our proposed novel context-aware vandalism detection technique is an exploratory step towards more complex classification techniques to detect progressively sneakier text vandalism on Wikipedia.

The next chapter continues the work in Chapter 6 to show the applicability of CLVD techniques in other domains of research. We show that text features used in detecting malicious editing activities (i.e. vandalism) on Wikipedia are also effective in predicting whether spam emails contain malicious attachments and (to a lesser extent) URLs. The implication of this research is the reduction in scanning emails for malicious content as the text of emails is a predictor of malicious intent.

Chapter 8

Predicting Malicious Content in Spam Emails

In this chapter, we show the applicability and extensibility of the vandalism (malicious content) detection techniques presented in Chapter 6 to detecting malicious content in spam emails. Malicious content in spam emails is increasing in the form of attachments and URLs. Malicious attachments and URLs attempt to deliver malicious software that can compromise the security of a computer. We show in chapter that the techniques from Chapter 6 can help reduce reliance on virus scanners and URL blacklists, which often do not adapt as quickly as the malicious content evolves.

This chapter has been published in Tran and Christen [2013a] with extensions for this thesis of a significantly larger data set and a new presentation of results (Section 2.7) consistent with the rest of this thesis. We begin by introducing the problem in Section 8.1, and in Section 8.2 review the spam email literature (specific to this chapter) to show how Wikipedia vandalism detection research provides solutions to a similar problem in a different domain. We formalise the problem of malicious spam emails in Section 8.3 and the real-world data sets that we used for our experimental evaluation in Section 8.4. In Section 8.5, we develop text features similar to those presented in Chapter 6 with additional text features and modifications to capture malicious intent in these spam data sets. Section 8.6 details our evaluation methodology and Section 8.7 summarises our results. We discuss our results in Section 8.8 and conclude our findings in Section 8.9.

8.1 Introduction

Email spam, unsolicited bulk email [Blanzieri and Bryl, 2008], accounted for an average of 66.5% of all emails sent in the first quarter of 2013, and of these 3.3% contained malicious attachments[1]. Estimates show that approximately 183 billion emails (i.e. 6 billion emails with malicious attachments) were sent every day in the first quarter

[1]Kaspersky Lab Securelist article: "Spam in Q1 2013." (8 May 2013) http://www.securelist.com/en/analysis/204792291/Spam_in_Q1_2013

of 2013[2]. Malicious attachments and embedded URLs (Universal Resource Locators – also known as Web links) are attempts to infect the computer of a recipient with malware (malicious software) such as viruses, trojans, and keyloggers. Malicious attachments in an email are attempts to deliver malware directly, whereas malicious URLs are indirect. Spam emails with malicious content (attachments or URLs) try to entice the recipient into opening an attachment or to click on a URL. Such spam emails have subject and content text that entices or alarms the recipient into acting on the disguised malicious content.

To find this type of harmful spam emails, scanning the attachments of emails and URLs with virus scanners or against blacklists can reveal their scope and the nature of the malicious content. However, scanning emails requires external resources that are computationally expensive and difficult to maintain [Ma et al., 2009a]. This method of identifying spam and other spam filtering methods aim to be highly responsive to changes in spamming techniques, but are not sufficiently flexible to handle variations in spam emails [Blanzieri and Bryl, 2008].

The task of identifying malicious content (attachments or URLs) in spam emails has been subject to limited published research. Our specific definition of malicious software includes only malware and so is different from research on classifying malicious emails by analysing URLs in their names [Le et al., 2011], auxiliary information [Ma et al., 2011], or JavaScript code [Likarish et al., 2009]. Our research should help identify one of the most harmful types of spam emails received.

In this chapter, we propose several potential novel features for predicting malicious attachments and URLs in spam emails. We hypothesise that spam emails with malicious attachments or URLs can be predicted using only the text content in the email subject and body. Our work also differs from related work as it is self-contained (i.e. does not require external resources such as blacklists) and does not add risks of exposure to malicious content by attempts to analyse or scan dubious attachments, or by tracking URLs. The success of these malicious spam emails rely on crafted emails and targeted users, where only a person opening the malware can cause significant damage [Lee, 2012; Straight, 2014]. Lee [2012] shows a study of malicious emails that are being crafted to targeted specific users and their complexity allows them to bypass spam filters even with malware in their attachments. Straight [2014] highlights preventative measures for cyber security in general, but an example shows that malicious emails do get through and only person is needed to open the malware to cause significant damage to their workplace.

We use three real world data sets obtained from three different sources. The first data set is from the Habul plugin for the Thunderbird mail client[3], the second data set (named Botnet) is collected from honeypots around the world to study the characteristics of email spam botnets (see Section 8.4), and the third data set (named UserRep) is compiled from user reported spam emails from an email service provider.

[2]Radicati Group Reports – Executive Summary: "Email Statistics Report, 2013-2017." (22 April 2013) http://www.radicati.com/wp/wp-content/uploads/2013/04/Email-Statistics-Report-2013-2017-Executive-Summary.pdf

[3]https://addons.mozilla.org/en-us/thunderbird/addon/habul/

We extract specific features from the metadata and text content of these real world spam emails. The proposed features are self-contained (no need to scan emails using external resources such as virus scanners and blacklists); robust (high adaptability to changes in spamming techniques); and time efficient (process many emails per second). We apply a Random Forest (RF) classifier to these selected features to show their effectiveness in distinguishing risky spam emails (i.e. those with malicious attachments) from those without malicious attachments. However, these selected features are insufficient to comprehensively classify the spam emails into at risk or not (i.e. that is spam or without malicious URLs). We discuss the success and failure of our features in identifying malware associated with spam and the potential research directions that arise from this work.

Our contributions in this work are (1) developing new or novel features that do not require external resources for the task of classifying malicious spam emails, (2) evaluating these features on three real-world data sets, and (3) demonstrating how well malicious attachments can be predicted from only the content of the email itself with high classification scores. Our work aims to reduce the need to scan emails for malicious content to save time and resources.

8.2 Related Work

We summarise related work in respect to four aspects of our work, highlighting text and machine learning based approaches. We look at spam filtering and specifically related work on classifying malicious attachments and URLs.

Email Spam Filtering. Spam filtering is a well-developed field with many different techniques applied to many types of spam. A survey of machine learning based approaches to spam filtering by Blanzieri and Bryl [2008] covers ambiguous definitions of spam, summarises a variety of spam detection methods and their applicability to different parts of an email, and summarises the various data sets used in this research. The survey shows a variety of machine learning approaches for features extracted from the email header, body, and the whole email message. In particular, Hao et al. [2009] develops a lightweight reputation method based on features derived from the metadata of emails to determine their legitimacy based on the sender. This reputation method can be used as a first-pass filter for blacklists to improve classification accuracy of spam emails.

In summary, email spam filtering is a mature research field as shown by Blanzieri and Bryl [2008] with many filtering techniques available such as rule based, information retrieval based, machine learning based, graph based, and hybrid techniques. However, identifying emails with malicious content remains a problem worthy of further investigation.

Classification of Malicious Attachments. Emails containing malicious attachments are potentially one of the most harmful types of emails as the malware has the potential to do significant damage to computers and to spread rapidly [Basaras et al., 2013; Hofmeyr et al., 2013]. The user's email usage behaviour can also change

depending on the malware's capability for spreading infection. By engineering features that capture behavioural properties of email use and the content of emails, the outgoing email behaviour of users (i.e. what and when emails are sent) can predict when malware has compromised a computer [Martin et al., 2005]. Applying feature reduction techniques can further improve the classification accuracy of malware propagated in outgoing mail [Masud et al., 2007]. These approaches aim to identify new malware by observing behaviour after infection.

For preventative solutions that do not need to scan attachments, analysing properties of the software executables can reveal malicious intent [Wang et al., 2007]. Our work also aims to be preventative, but without adding the risk of infection by analysing software executables which may escape.

Classification of Malicious URLs. Research on classifying URLs for malicious intent extends beyond spam emails, because of the common nature of URLs in many Web documents and electronic messages. Blacklisting is a highly efficient method of preventing access to malicious URLs, but it relies on discovering which URLs are malicious beforehand [Ma et al., 2011]. Proactive methods of blacklisting that make predictions based on the DNS records of known bad URLs [Felegyhazi et al., 2010]. However, blacklisting services cannot keep up with high volume spamming botnets that operate from frequently changing URLs and IP addresses [Ramachandran et al., 2006; Eshete and Villafiorita, 2013].

To be fully effective and adaptive to new malicious URLs, creating relevant URL features or variables based on text and hosting properties for classifiers has been shown to be effective in predicting maliciousness of URLs [Ma et al., 2009a; Le et al., 2011; Khami et al., 2014]. However, these features require many external resources such as IP blacklists, domain registration details, DNS records, and reliable geographical location of IP addresses [Khami et al., 2014]. Although these features can be applied in the real-time classification of URLs, there are trade-offs in accuracy and processing time [Ma et al., 2009b].

Other methods for the detection of malicious URLs require access to the Web pages of URLs and then performing further analysis [Khami et al., 2014]. Parts of Web pages can be obfuscated to hide malicious intent, such as malicious Javascript code [Likarish et al., 2009; Khami et al., 2014]. By developing many different sets of features or variables over both the structure and content of a Web page, a comprehensive analysis can be performed to determine the likelihood that a Web page is malicious [Canali et al., 2011; Eshete and Villafiorita, 2013].

In this work, we do not consider all the possible features for classifying URLs as discussed by other researchers. Our focus is on using email content alone to predict if a URL is malicious. In future work, we intend to perform further analysis of these promising features used in related work, and apply (and where possible improve) them in our identification of risky emails.

Wikipedia Vandalism Detection. In this work, we borrow some text features from the related field of vandalism detection on Wikipedia as discussed in Chapters 6 to 7. The problem of vandalism detection (i.e. a malicious edit or change in the content) and detecting emails with malicious content are related and may share

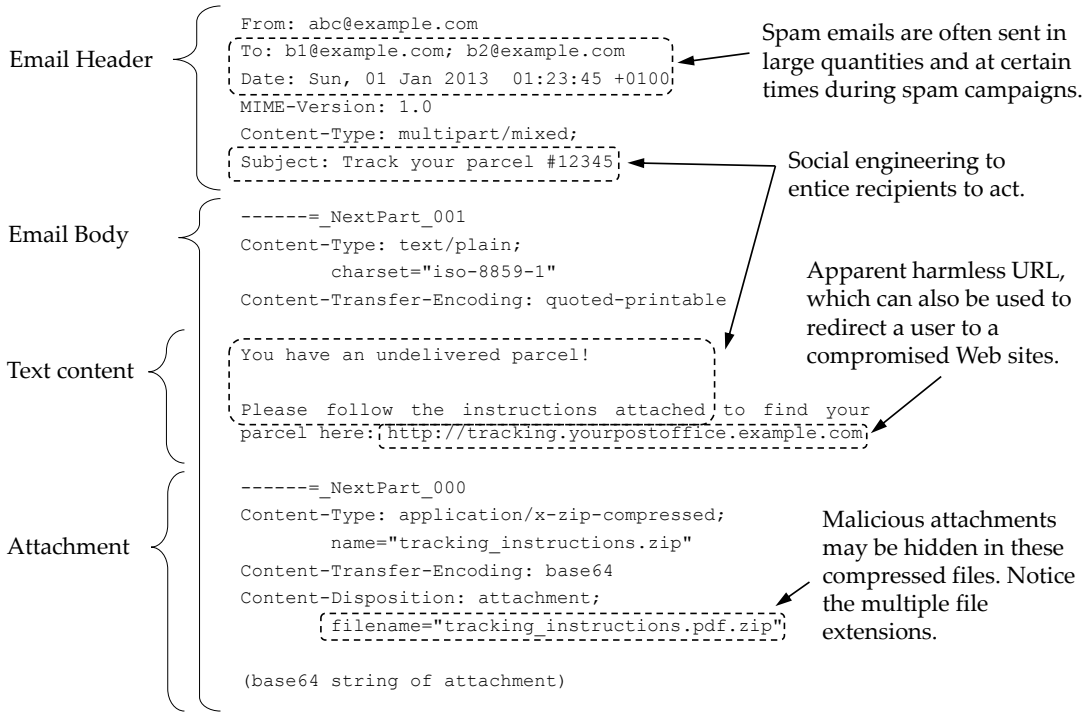

Figure 8.1: An example (fake) spam email with a potential malicious attachment and URL.

similar characteristics. The text within a Wikipedia article and text in an email may contain content that distinguishes it from a normal article or normal (spam) email, respectively. For example, abnormal use of vulgar words or excessive use of uppercase words may hint at malicious intent. Our work provides a comparison of the classification models across these two application areas, and also helps to address the problem of insufficient training samples for the testing of classification models.

The PAN Workshops in 2010 and 2011 (Section 2.5) held competitions for vandalism detection in Wikipedia, where they released a data set containing manually classified cases of vandalism. In Section 8.7, we describe our selected text features from the winners of the competitions in 2010 [Mola-Velasco, 2010] and 2011 [West and Lee, 2011], and additional features relevant to this work. These text features aim to show text regularities within spam emails.

8.3 Malicious Spam Emails

Spam emails vary from annoying, but harmless advertising to harmful scams, fraudulent activity, and other cybercrime activities [Blanzieri and Bryl, 2008]. Spam emails with malware or URLs that direct users to malware are common methods used by cybercriminals to find new victims. For example, spammers may want to expand their botnets for phishing emails, or cybercriminals may use them to propagate their computer viruses to harvest passwords, credit cards, bank accounts, and other sensi-

tive personal information. Our work aims to provide a preventative method to help reduce the propagation of malware using spam emails. Before presenting our results, we briefly describe our raw data of malicious spam emails and how cybercriminals disseminate spam emails.

Email formats are well-known, but less familiar are the raw email data that we use to construct our features. We present an example of a (fake) spam email with potential malicious content in Figure 8.1, stripped of irrelevant metadata. The figure shows an email in raw text format with annotations showing important parts of the email that can be used for the construction of features/variables. We have the email header that contains delivery instructions for mail servers, and the email body that can have many sections for text, attachments, and other types of attachable data. Emails are identified as spam in two ways: a user determines if an email is spam, and emails collected and identified as sourced from known spamming networks. Both scenarios for determining spam are captured in our real world data sets described in the following section.

Our example in Figure 8.1 shows the typical structure of a malicious spam email. The subject or text content of such emails often contains social engineering methods to manipulate recipients into first reading and then act on the email. In this case, we have the premise of a fake undelivered parcel that requires the recipient to download a compressed file (purposefully misleading with multiple file extensions). This compressed file serves the purpose of hiding malware executables from virus scanners operated/applied by mail servers. The URL in this example acts as a secondary method of delivering malicious content. Similar to attachments, malicious URLs can disguise a malicious Web site (e.g. example.com) by adding subdomains representing a known and safe Web site (e.g. tracking.yourpostoffice). Our example also shows a possible spam template, where attachments or URLs may have different names, but the same malicious intent.

Spam templates are often used in spam campaigns, where many emails are sent in a short period of time often with minor lexical variations to their content [Stone-Gross et al., 2011]. In our example in Figure 8.1, variations can occur in the tracking number, attachment name, and URL. These variations are attempts to prevent basic routine spam detection methods applied by mail servers. Other obfuscation methods include manipulation of email headers to include legitimate email addresses that also help avoid spam filtering and thus allow more spam emails to be sent undetected.

The emergence and proliferation of botnets have allowed large quantities of spam emails to be sent in a coordinated way, and amplify cybercrime activities [Broadhurst et al., 2013]. Botnets are networks of compromised computers controlled by a 'botmaster' who often rents the botnet to spammers and others that intent to use them to deliver malware. Botnets are the backbone of spam delivery, and estimates suggest that approximately 85% of the world's spam email were sent by botnets every day [John et al., 2009]. The widespread use of botnets shows how spammers understand and manipulate the networks of compromised computers and servers around the world to ensure high volumes of spam are delivered to large numbers of Internet users.

Table 8.1: Habul data set statistics for 2012.

Habul Emails		with Attachments		with URLs	
Month	Total	Total	Malicious	Total	Malicious
Jan	67	7	(43%) 3	25	(12%) 3
Feb	104	10	(20%) 2	33	(18%) 6
Mar	75	5	(0%) 0	28	(14%) 4
Apr	65	4	(50%) 2	26	(8%) 2
May	83	4	(0%) 0	38	(13%) 5
Jun	94	1	(0%) 0	41	(12%) 5
Jul	72	2	(50%) 1	26	(42%) 11
Aug	85	0	(0%) 0	46	(22%) 10
Sep	363	11	(64%) 7	140	(3%) 4
Oct	73	1	(100%) 1	11	(27%) 3
Nov	193	4	(0%) 0	89	(15%) 13
Dec	95	6	(50%) 3	31	(39%) 12
Total	1,369	55	(35%) 19	534	(15%) 78

Overall, the use of spam emails is an important vector to propagate malware, and the forms of social engineering used in spam emails have grown more sophisticated, improving the ability to deceive many users into malware self-infection.

8.4 Email Spam Data Sets

We use three real world data sets compiled in 2012 from three different spam collection sources, which were all obtained through different confidentiality agreements. Further description and criminal analysis of these data sets can be found in Alazab and Broadhurst [2014]. The first data set is compiled from the Habul Plugin for Thunderbird [4] (an email client) that uses an adaptive filter to learn from a user's labelling of emails as spam or normal to automatically classify future incoming emails. Table 8.1 summarises the statistics for the Habul data set, which are compiled monthly. The second data set is compiled from a global system of spam traps designed to monitor information about spam and other malicious activities. The second data set we label as the Botnet data set, which were also compiled monthly. Table 8.2 summarises the descriptive statistics for our larger Botnet data set. The third and final data set we label as "UserRep" to signify that it was compiled from user reported spam emails from an email service provider (who has requested to remain anonymous). The UserRep data set is also compiled monthly, where Table 8.3 shows the data set statistics. We received all data sets in anonymised form, so no identifiable email addresses or IPs are available for analysis.

For each email, we extract attachments and URLs and upload these to VirusTotal[5],

[4] https://addons.mozilla.org/en-us/thunderbird/addon/habul/
[5] https://www.virustotal.com/en/

Table 8.2: Botnet data set statistics for 2012.

Botnet Emails		with Attachments		with URLs	
Month	Total	Total	Malicious	Total	Malicious
Jan	31,991	141	(19%) 27	12,480	(0.03%) 4
Feb	49,085	534	(12%) 66	14,748	(0.03%) 4
Mar	45,413	542	(10%) 52	19,895	(0.12%) 23
Apr	33,311	330	(53%) 175	12,339	(0%) 0
May	28,415	756	(78%) 592	13,645	(0.02%) 3
Jun	11,587	102	(55%) 56	8,052	(1%) 80
Jul	16,251	442	(44%) 196	5,615	(2%) 92
Aug	21,970	297	(38%) 113	16,970	(4%) 707
Sep	27,819	290	(4%) 12	17,924	(2%) 442
Oct	13,426	904	(58%) 524	4,949	(0.04%) 2
Nov	17,145	1,113	(79%) 882	7,877	(1%) 49
Dec	20,696	634	(49%) 313	7,992	(3%) 241
Total	317,109	6,085	(49%) 3,008	142,486	(1%) 1,647

a free online virus checker that offers support for academic researchers, to scan for viruses and suspicious content. VirusTotal uses over 40 different virus scanners, where we consider an attachment or URL to be malicious if at least one scanner shows a positive result. For this study, we only focus on emails with attachments or that contains URLs to predict/identify emails with malicious content.

The Habul data set is much smaller than the Botnet data set and the UserRep data set, but has the advantage that these emails have been manually labelled as spam by recipients. This means the spam in the Habul data set has been viewed, but the Botnet data set contains spam that circulated all over the world, but without the certainty that the emails have reached their intended targets. The UserRep data set is significantly larger than both the Habul and Botnet data sets, which suggests a large user population labelling emails as spam. We see a large and consistent presence of malicious spam in the UserRep data set, especially with malicious URLs.

All of the data sets show some similarities: nearly half of spam emails contain at least one URL, but only a relatively small percentage was identified as malicious. In contrast, many more emails that include an attachment were malicious. For each data set, there were peaks of spam during 2012 that either contained malicious content or not, and which suggested different types of spam (mass propagation) campaigns. These campaigns usually shared similarities in the content of their emails, and this alone may indicate the risk of malicious content.

8.5 Feature Engineering

We now explore a comprehensive set of features that help characterise email content. We borrow some of these features, as noted, from the related field of vandalism detection on Wikipedia, which are detailed in Chapter 6. The aim of vandalism

Table 8.3: UserRep data set statistics for 2012.

UserRep Emails		with Attachments		with URLs	
Month	Total	Total	Malicious	Total	Malicious
Jan	17,512	944	(10%) 99	6,793	(21%) 1,398
Feb	867,725	86,491	(45%) 38,556	381,304	(22%) 83,327
Mar	297,999	19,307	(25%) 4,769	129,051	(21%) 27,511
Apr	1,026,077	62,019	(28%) 17,407	405,476	(19%) 75,801
May	1,084,214	59,754	(25%) 14,675	430,242	(20%) 87,699
Jun	1,521,541	32,667	(16%) 5,136	594,092	(18%) 104,080
Jul	1,710,530	36,558	(11%) 3,859	744,368	(35%) 260,084
Aug	1,102,896	42,764	(9%) 3,724	552,584	(19%) 104,476
Sep	1,069,399	47,744	(9%) 4,145	539,869	(15%) 80,276
Oct	937,430	42,966	(10%) 4,312	418,755	(16%) 68,399
Nov	1,194,951	34,164	(9%) 3,175	674,579	(20%) 136,686
Dec	2,301,783	21,607	(11%) 2,382	1,210,647	(29%) 356,225
Total	13,132,057	486,985	(21%) 102,239	6,087,760	(23%) 1,385,962

detection is to identify malicious modifications to articles. In particular, we borrow some text features from the winners of vandalism competitions held at the PAN Workshops in 2010 and 2011 [Mola-Velasco, 2010; West and Lee, 2011]. As far as we are aware, none of the features described below have been used to predict malicious content in emails.

We describe the novelty of these features, which we use as risk variables, in the context of their applications in related areas of research. Features in bold text are novel features not used in Wikipedia vandalism detection nor for malicious spam email detection. We use a standard feature set for the email content, and different sets of features for the attachments and URLs because they are inherently different types of data with few shared attributes. This is evident from the attachment files and URL text, and the diverging research in cyber-security into malware detection and malicious URL detection as discussed in Section 8.2.

8.5.1 Feature Description

Table 8.4 shows our features and a summary description. Features with prefix `H` are email header features; prefix `S` are subject features; prefix `P` are payload features (or content of email); prefix `A` are features of attachments; and prefix `U` are features of URLs. We describe these features in detail below and how these groups of features are related. These features (and experiments) were generated on an Intel® Core™ i7-2600 CPU @ 3.40GHz, with 16GB RAM, and 2TB 7200rpm mechanical hard drive.

- **Header Features**. These features are extracted from the metadata of emails. Since the emails have been anonymised, we can create only limited number of features.

Table 8.4: Email features used in experiments. Features in bold text are novel features not seen in other research areas, or specific to malicious spam email detection.

Feature	Description
H01-DAY	Day of week when email was sent.
H02-HOUR	Hour of day when email was sent.
H03-MIN	Minute of hour when email was sent.
H04-SEC	Second of minute when email was sent.
H05-FROM	Number of "from" email addresses, known as email senders.
H06-TO	Number of "to" email addresses, known as email recipients.
S01-LEN	Number of characters.
S02-PW	Number of pronoun words.
S03-VW	Number of vulgar words.
S04-SW	Number of slang words.
S05-CW	Number of capitalised words.
S06-UW	Number of words in all uppercase.
S07-DW	Number of words that are digits.
S08-LW	Number of words containing only letters.
S09-LNW	Number of words containing letters and numbers.
S10-SL	Number of words that are single letters.
S11-SD	Number of words that are single digits.
S12-SC	Number of words that are single characters.
S13-UL	Max of ratio of uppercase letters to lowercase letters of each word.
S14-UA	Max of ratio of uppercase letters to all characters of each word.
S15-DA	Max of ratio of digit characters to all characters of each word.
S16-NAA	Max of ratio of non-alphanumeric characters to all characters of each word.
S17-CD	Min of character diversity of each word.
S18-LRC	Max of the longest repeating character.
S19-LZW	Min of the compression ratio for the lzw compressor.
S20-ZLIB	Min of the compression ratio for the zlib compressor.
S21-BZ2	Min of the compression ratio for the bz2 compressor.
S22-CL	Max of the character lengths of words.
S23-SCL	Sum of all the character lengths of words.
P01 to P12, **P13 to P23**	Same as features S01 to S23, but for the email payload (content).
A01-UFILES	Number of unique attachment files in an email.
A02-NFILES	Number of all attachment files in an email.
A03-UCONT	Number of unique content types of attachment files in an email.
A04-NCONT	Number of all content types of attachment files in an email.
U01-UURLS	The number of unique URLs in an email.
U02-NURLS	The number of all URLs in an email.

  - Features **H01-DAY** to **H04-SEC** are simple variables that capture the time when emails were sent. Emails propagated via a spam campaign are often sent at the same time en masse from multiple servers [Stone-Gross et al., 2011]. Due to the anonymisation of emails and non-disclosure of sources of spam emails because of confidentiality reasons, we could only normalise the sent times of emails to Greenwich Median Time (GMT) instead of local timezones, server times, or other localisation methods.
  - Features H05-FROM and H06-TO are counts of the email addresses of the sender and intended recipients. Since these features have been anonymised, we only count the number of addresses. Further analysis of these email addresses is warranted, especially if addresses are at least partially revealed, because it is likely that more detailed features will help identify particular spam campaigns.

- **Text Features**. These features are applied to the subject (prefix S) and payload (prefix P) of emails. Although we apply these features identically on different data, they require different interpretation for subject and payload data. For text in the subject and payload, we extract a list of words and then count the number of appearances of each word.

  - Feature S01-LEN (P01-LEN) is a simple count of the number of characters in the text of the subject or payload.
  - Features S02-PW to S04-SW (P02-PW to P04-SW) are a count of special words in malicious emails. We obtained lists of these words from the English Wiktionary[6] and applied them to both data sets. This word mapping produced 27 unique pronoun words, 1064 unique vulgar words, and 5,980 unique slang words. The presence of these word form features were strong indicators of a spam email and also of possible malicious content especially when the 'payload' attempted to persuade users to download files or follow a URL. These words features were borrowed from Chapter 6 and directly from the PAN Workshops [Mola-Velasco, 2010; West and Lee, 2011], but we used different sources to identify these words.
  - Features S05-CW to S12-SC (P05-CW to P12-SC) are also borrowed from the PAN Workshops [Mola-Velasco, 2010; West and Lee, 2011]. These features are self-descriptive and look for patterns in the words used in the subject and payload of emails. We expect these features to distinguish genuine emails from spam campaigns because such campaigns often use email text templates [Kreibich et al., 2009].
  - Features **S13-UL** to **S23-SCL** (**P13-UL** to **P23-SCL**) are our set of new features. These features look closely at the distribution of character types in the form of ratios. We select out the maximum and minimum of each features applied to each word to highlight any unique oddities in the words

[6] http://www.wiktionary.org/

used in the email subject and payload. Our definitions of two of the less self-descriptive features are as follows:

* Character diversity is from Mola-Velasco [2010] and interpreted here as a measure of the number of different characters in a word compared to the word length: $length(word)^{\frac{1}{set(\{character \in word\})}}$
* Compression ratio is defined as: $\frac{size_{uncompressed}}{size_{compressed}}$

In the subject of spam emails, these emphasise unique words much stronger than features `S02-HOUR` to `S12-SC`, because of the relatively shorter length of text to the payload.

– Features **`S18-LRC`** to **`S21-BZ2`** are variants of the same concept of identifying particular words with repetitive characters. We use these features to account for simple misspellings of words by repeating characters. These are the most computationally intensive features, with feature **`S19-LZW`** on average taking 4 milliseconds (ms) per email (machine details at the start of this section), and features **`S18-LRC`**, **`S20-ZLIB`**, and **`S21-BZ2`** on average taking less than 1 ms. All other features on average took between 0.005 ms and 0.010 ms per email. Note that these are the time taken to generate a single feature and does not include parallelisation and batch pre-processing of the required data.

- **Attachment Features**. These features (prefix `A`) are specific to spam emails with attachments. We do not use URL features with these attachment features. Our investigation looks only at simple, but novel, features of how attachments appear in emails. In particular, we count the number of files and the declared content types (such as image or zip files). For spam emails with attachments, malicious attachments may appear as the only attachment in emails, or hide in compressed files or with multiple extensions. In future work, we aim to generate more features from file names or other attributes of attachments and so hope to avoid the need to scan for malicious content.

- **URL Features**. These features (prefix `U`) are specific to spam emails with URLs. We do not use these features in conjunction with the attachment features, but they are novel to our classification task. In future work, we intend to apply more complex text analysis specifically for URLs in order to extract features that may distinguish URLs that are designed to direct users to websites (with and without malicious content). For example, this may occur when a number of URLs shares a common domain names or common access pages.

8.5.2 Feature Ranking

With many varieties of potential variables or features, we find features important to our classification task and compare them across the three data sets. We compare them by using the Random Forest classifier that produced a ranking of these features

Table 8.5: Top 5 features determined by a Random Forest classifier for the data split of November. Scores are the information entropy of features. Scores range from 0.0 to 1.0, where 1.0 means most information is gained when decision trees are split on this feature.

Data set	Habul		Botnet		UserRep	
Data type	Feature	Score	Feature	Score	Feature	Score
Attachments	S05-CW	0.112	**S21-BZ2**	0.107	**S13-UL**	0.071
	S23-SCL	0.081	**S20-ZLIB**	0.086	**H06-TO**	0.067
	S09-LNW	0.074	**S17-CD**	0.072	S05-CW	0.067
	S15-DA	0.067	**S19-LZW**	0.058	**S14-UA**	0.051
	H02-HOUR	0.063	**S22-CL**	0.045	**S23-SCL**	0.049
URLs	**U02-NURLS**	0.088	**H01-DAY**	0.063	**H01-DAY**	0.082
	U01-UURLS	0.072	P01-LEN	0.056	**U02-NURLS**	0.069
	P09-LNW	0.053	**P23-SCL**	0.054	P01-LEN	0.065
	P21-BZ2	0.051	**H03-MIN**	0.053	**P21-BZ2**	0.052
	P08-LW	0.041	**H02-HOUR**	0.048	**U01-UURLS**	0.045

based on their entropy scores [Pedregosa et al., 2011]. See Section 8.7 below for a description of our classifier and classification results.

The entropy score measures the additional information gained when a decision tree (in the forest) is split on that feature. The aim is to have the most homogeneous decision branches after a split to improve classification scores. For example, for emails with attachments in the Botnet data set, we gain more than twice as much information by splitting on feature S21-BZ2 (0.107) than when we split on the feature S22-CL (0.045). To account for the overall randomness of the Random Forest classifier, we present the average scores of 10 training iterations in Table 8.5 for the data split of the month of November in both data sets (details below in Section 8.6). We bold features that are our novel contributions.

From Table 8.5, we see the majority of the best performing features are our proposed features for this classification task. In particular, for the larger Botnet data set with a larger number of emails, we find our selected features perform consistently well. The variety of features shows that no single feature dominates among the top 5 scores across all three data sets, and attachments and URLs. This result further emphasised the need for a feature rich model to capture variations in different types of spam emails containing malicious content.

For the Habul data set, predicting malicious attachments and URLs from email content shows different but also important features. For attachments, we find features S05-CW, **S23-SCL**, S09-LNW, and **S15-DA**, suggested emails with capitalised words containing letters and digits in the subject line. This apparent formality in the subject line attempts to mimic legitimacy in order to gain the trust of recipients to open the email and download the attachments. The presence of feature **H02** also suggests these malicious spam email may be originating from a spam campaign. For URLs, we find URL and payload features are relevant when **U02-NURLS** and **U01-UURLS** appear together perhaps indicative that a few unique URLs are at risk.

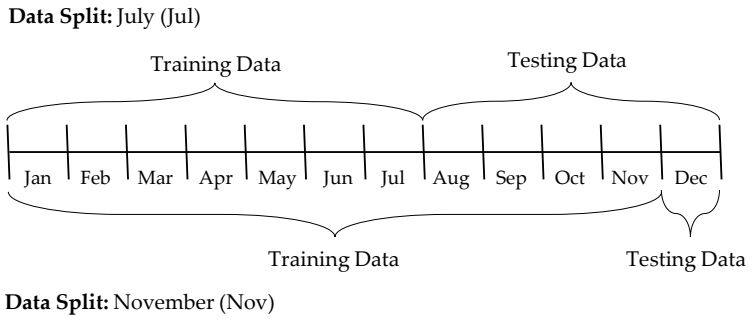

Figure 8.2: Illustration of splitting data into training and testing sets.

This suggests malicious spam emails contain fewer URLs with associated content designed to persuade recipients to click on those URLs.

For the Botnet data set, we find the subject of the email to be the strongest predictor of the presence of malicious attachments, whereas when the email was sent was a good predictor of malicious URLs. For attachments, we found the email subjects with low compressibility of words for all three compression algorithms (**`S19-LZW`**, **`S20-ZLIB`**, and **`S21-BZ2`**), combined with many different characters (**`S17-CD`**), and long words (**`S22-CL`**) were also useful predictors. This suggested subject lines with seemingly random characters, which may trigger the recipient's curiosity to download the (malicious or risky) attachments associated with the email. For URLs, the time features are highly predictive along with the length of the content of the email. Again this indicates spam campaigns with email templates that offer strange or unconventional subject text may induce the curiosity of recipients to download the associated attachment(s).

We find that for each data set, the set of important features for identifying malicious attachments or URLs are very different. This suggests a wide variety of features are needed for malicious spam emails from different sources. Some similarities in important features show that emails with attachments indicate their likely malicious intent by their subject line; and for those emails with URLs, the frequency or number of URLs, the text, and the time when the emails were sent were predictive of malicious intent of URLs.

8.6 Evaluation Methodology

As our data sets are already partitioned into months, we combine the months progressively to learn on the earlier months and test our classifier on the later months. Figure 8.2 illustrates our data splitting process into training and testing data sets for the months of July and November. For example, for the months of July, we train on all spam emails with malicious content from January to July, and then test the model on spam emails with attachments or URLs from August to December. This shows the

effects of different training sample sizes on classification quality, and the adaptability of the classifiers used.

We combine the feature sets differently for classification of attachments and URLs. For attachments, we choose features with the prefixes of `H`, `S`, `P`, and `A`. For URLs, we choose features with prefixes of `H`, `S`, `P`, and `U`. These separate groups of features allow us to target for malicious attachments or malicious URLs separately in our evaluation. In future work, we could expand on each feature sets as needed to improve classification results when new trends emerge or are discovered.

We use three classifiers to evaluate our features: Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM); and we use the evaluation metrics from the Scikit-learn toolkit [Pedregosa et al., 2011]. The NB and SVM classifiers are commonly used in spam classification [Blanzieri and Bryl, 2008]. We include the RF classifier in our evaluation as it has shown good performance with the features from Chapter 6. We performed a standard grid search with 10-fold cross validation to determine the best parameters for each classifier. Our evaluation measures are explained in Section 2.7, where the positive class is the spam emails with malicious content and the negative class is the spam emails with non-malicious content.

To the best of our knowledge, we are the first to use these methods to predict malicious content in emails. There are no comparable baseline measures available for comparison. In future work, we plan to expand our set of URL features and compare these to related work on the prediction of phishing URLs in emails. For now, we present our classification results and discuss our findings.

8.7 Classification Results

We compare the classification results for the three classifiers in Figures 8.3 to 8.14 for AUC-PR and AUC-PR scores. In Figures 8.15 to 8.18, we compare our classification results for the RF classifier. We compare the data splits in each figure for three different data sets and three different classifiers. Our figures also show the effect of the accumulation each month of the spam data on predicting malicious emails in the subsequent months.

For spam emails with attachments, predicting whether attachments are malicious is successful on the Botnet data set, reaching a peak AUC-PR score of 0.952 (Figure 8.5) and similarly high peak AUC-ROC score of 0.914 (Figure 8.6). The low AUC-PR score for the training set split in January was expected as we have insufficient data to observe whether attachments are malicious in the subsequent months (February to December). The classifier shows very poor performance on the Habul data set for many data splits (Figures 8.3 and 8.4). The reason is clear from Table 8.1, where we see again very few emails with attachments for the classifier to learn from. In some months corresponding with the other data splits (e.g. August), we do not have any or few emails with malicious attachments to learn from. The UserRep data set also shows poor AUC-PR scores (Figure 8.7), but consistent AUC-ROC scores (Figure 8.8). These results suggest that the spam emails collected from the Botnet

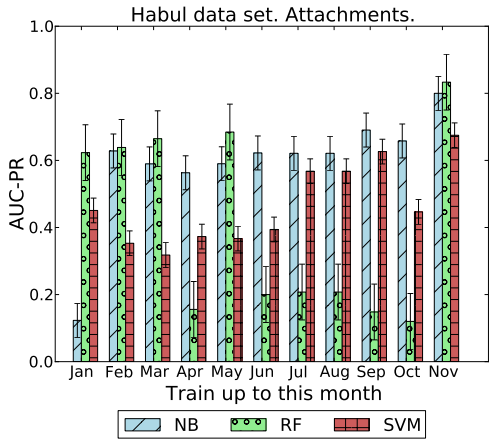

Figure 8.3: Habul data set. AUC-PR scores for detecting malicious attachments. Error bars show one standard error.

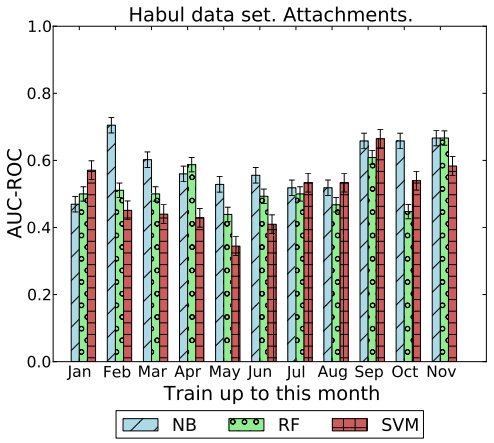

Figure 8.4: Habul data set. AUC-ROC scores for detecting malicious attachments. Error bars show one standard error.

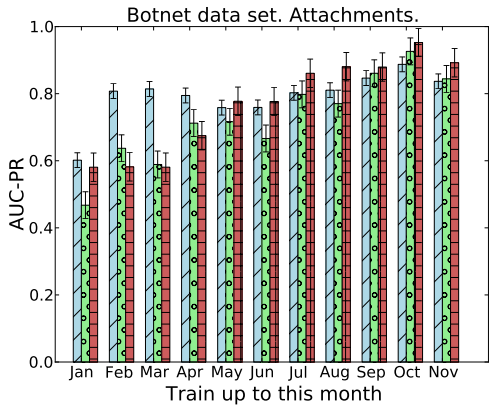

Figure 8.5: Botnet data set. AUC-PR scores for detecting malicious attachments. Error bars show one standard error.

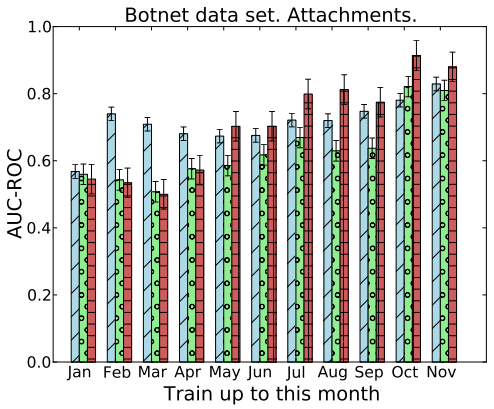

Figure 8.6: Botnet data set. AUC-ROC scores for detecting malicious attachments. Error bars show one standard error.

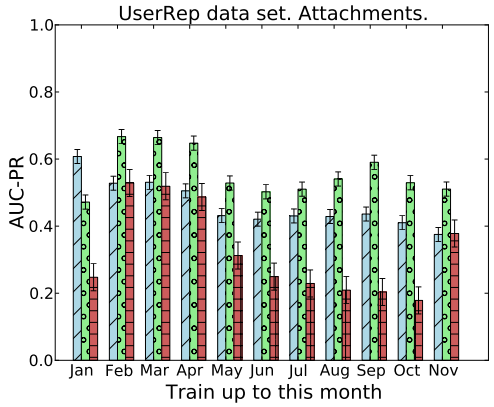

Figure 8.7: UserRep data set. AUC-PR scores for detecting malicious attachments. Error bars show one standard error.

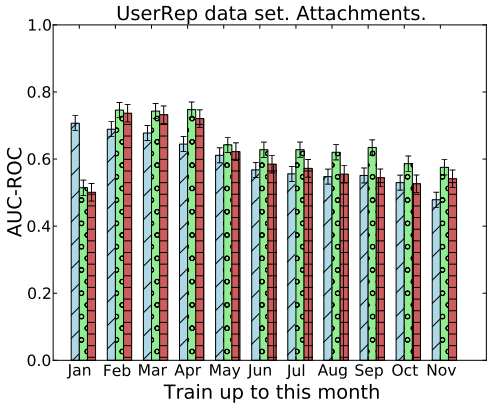

Figure 8.8: UserRep data set. AUC-ROC scores for detecting malicious attachments. Error bars show one standard error.

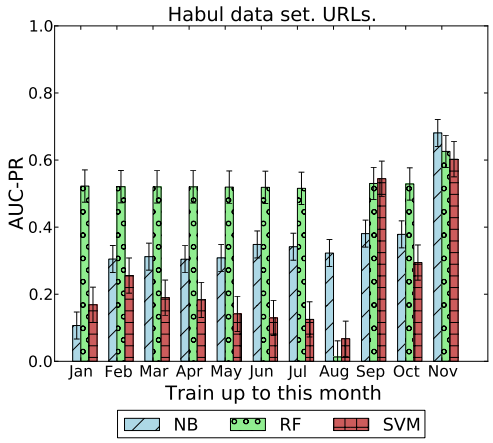

Figure 8.9: Habul data set. AUC-PR scores for detecting malicious URLs. Error bars show one standard error.

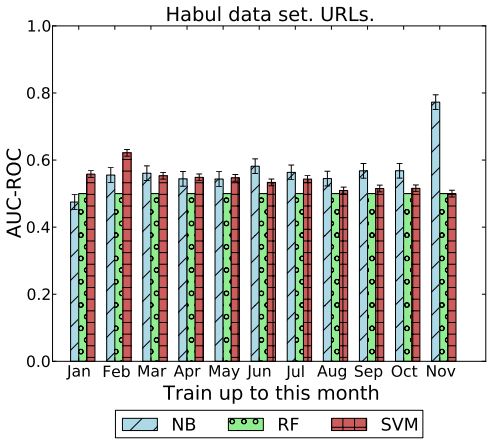

Figure 8.10: Habul data set. AUC-ROC scores for detecting malicious URLs. Error bars show one standard error.

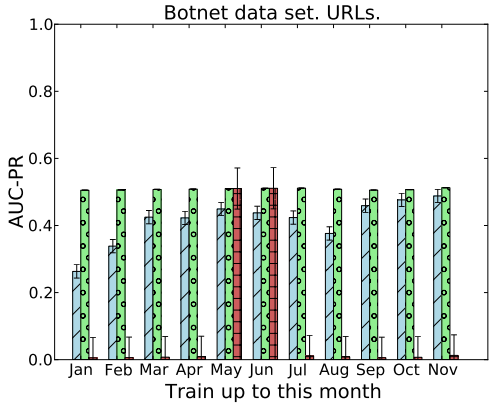

Figure 8.11: Botnet data set. AUC-PR scores for detecting malicious URLs. Error bars show one standard error.

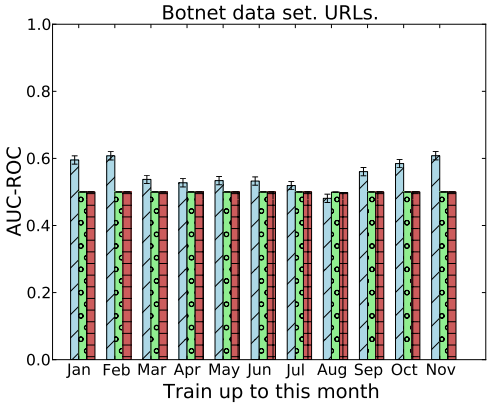

Figure 8.12: Botnet data set. AUC-ROC scores for detecting malicious URLs. Error bars show one standard error.

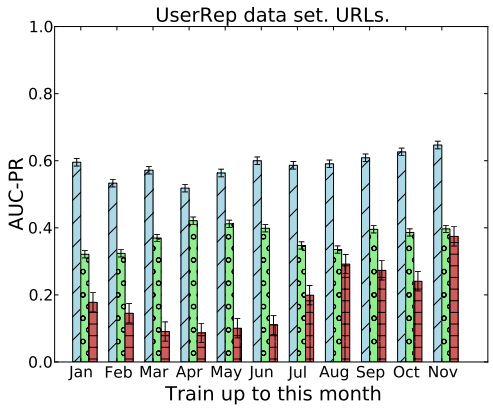

Figure 8.13: UserRep data set. AUC-PR scores for detecting malicious URLs. Error bars show one standard error.

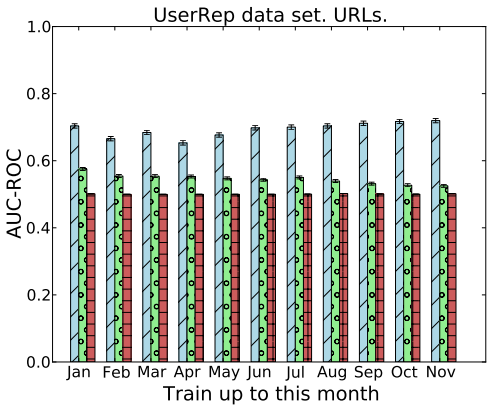

Figure 8.14: UserRep data set. AUC-ROC scores for detecting malicious URLs. Error bars show one standard error.

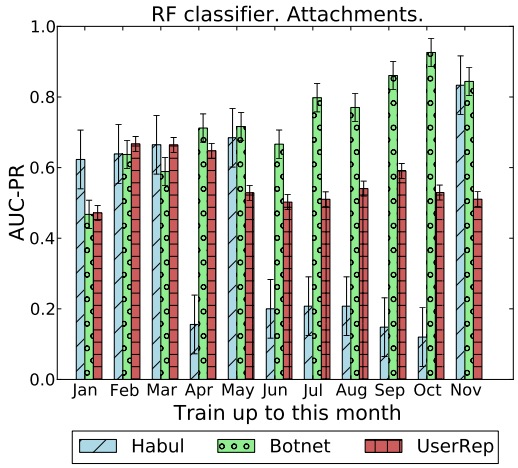

Figure 8.15: Comparison of AUC-PR scores of all data sets for the RF classifier in detecting malicious attachments. Error bars show one standard error.

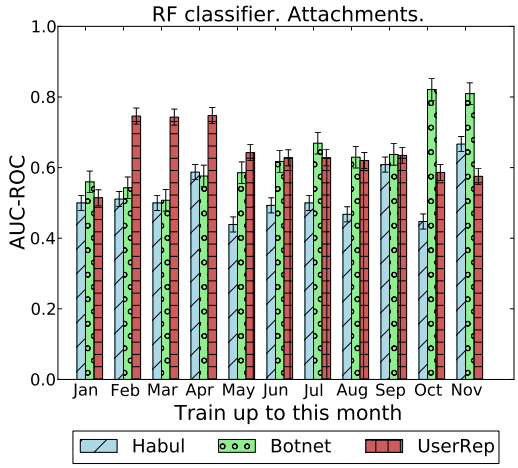

Figure 8.16: Comparison of AUC-ROC scores of all data sets for the RF classifier in detecting malicious attachments. Error bars show one standard error.

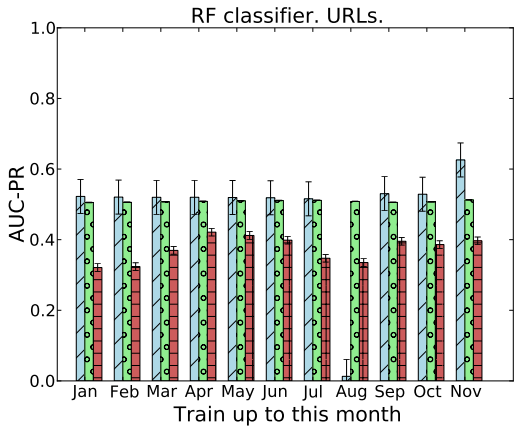

Figure 8.17: Comparison of AUC-PR scores of all data sets for the RF classifier in detecting malicious URLs. Error bars show one standard error.

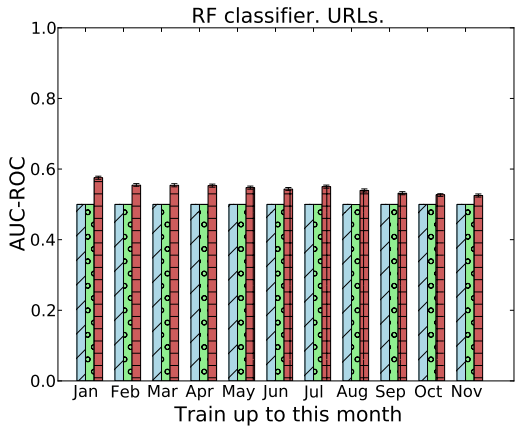

Figure 8.18: Comparison of AUC-ROC scores of all data sets for the RF classifier in detecting malicious URLs. Error bars show one standard error.

honeypots have similar characteristics that is predictive of maliciousness of their attachments, whereas user reported spam emails do not have comparatively clear text markers of malicious attachments.

For spam emails with URLs, all three classifiers show poor performance with AUC-PR scores (Figures 8.9, 8.11, and 8.13) around or below 0.5. This means for an email with malicious URLs, the classifiers NB and SVM will label them correctly less than 50% (or 0.5) of the time for the Habul and Botnet data sets. This is in contrast to a randomly selected email being classified as malicious, which is shown by the AUC-ROC score.

The NB classifier has reasonably good AUC-PR (Figure 8.13) scores for the UserRep data set. The AUC-ROC scores for predicting malicious URLs (Figures 8.10, 8.12,

and 8.14) have some good results, in particular the NB classifier, which shows scores more than 0.7 in the UserRep data set and some data splits in other data sets. Overall, the AUC-ROC scores for all classifiers on all data sets do not fall below 0.5, which means malicious URLs in spam emails can be predicted better than a random guess for most months in all data sets.

In Figures 8.15 to 8.18, we compare the classification results between our three data sets for the generally robust classifier: Random Forest (RF). As discussed above, the comparatively numerous training samples in the Botnet data set allow for a high classification performance as measured for both AUC-PR and AUC-ROC scores. The data split of November which has the most training samples also showed high classification scores, especially in the Habul data set, where there are fewer data samples. The figures show significant differences between the classification scores for predicting the maliciousness of attachments, where scores are consistently high for the Botnet data set. However, for URLs, the features we chose are not sufficient for the RF classifier to distinguish malicious URLs. An exception is the NB classifier, which shows high scores for predicting malicious URLs from just the email text.

Overall, our work shows the viability of predicting whether attachments and URLs in spam emails are malicious. Our proposed feature-rich model shows our hypothesis is true for malicious attachments as those emails can be predicted from the email subject and payload with high AUC-PR and AUC-ROC scores. In future work, we aim to look at adding more features for URLs, focusing on the lexical content (as in related work) to avoid drawing on external resources, such as blacklists. Our success with predicting malicious attachments reduced the need to scan every attachments for malicious content. When the data set is large and has a sufficient number of risky emails, we can reduce the need to scan over 95% of (spam) emails with attachments (from AUC-PR scores) by analysing only the text in emails with attachments.

8.8 Discussion

These findings are encouraging as they suggest we may be able to correctly identify over 95% of the 6 billion emails with malicious attachments sent everyday (see Section 8.1) by analysing only the email subject and text content. While our success was not as high when identifying malicious URLs, our results for attachments (Figures 8.7 and 8.8) show spam emails collected from botnets and honeypots (Botnet data set) have text patterns that allow the prediction of the maliciousness of their attachments.

The main advantage of our approach is the self-contained sets of features extracted from only the email itself may be sufficient to identify risky email, without recourse to external resources such as virus scanners or blacklists. This means our machine learning algorithms can quickly adapt to changes and evolution of spam emails, where the correctness of learning models can be verified and updated when scanners and blacklists have been updated.

A limitation of our approach is the descriptiveness of our proposed sets of fea-

tures. Our results show that the features are more suitable for predicting malicious attachments than malicious URLs. This suggests emails with malicious URLs do not have sufficient commonalities when the subject or text content are used to predict the malicious intent of its URLs. Some exploit kits such as the Blackhole Exploit Kit[7] simply inserts malicious URLs into emails without changing their content [Oliver et al., 2012]. Thus, non-malicious spam emails can become, via this method, malicious without any changes to their original spam content. To resolve this limitation, in future work we intend to add lexical features from related work (see Section 8.2) to our own tests for the risk of malware embedded in URLs, and compare their classification performance.

Another limitation is the possibility that a few spam campaigns have been over-represented in our data sets. We have not yet performed a detailed spam campaign analysis and this would be another research topic worth following up. Reviewing statistics from Tables 8.1, 8.2, and 8.3; for the Habul data set, we found 13 unique malicious attachments (in 19 emails with malicious attachments), and 73 unique malicious URLs (in 78 emails with malicious URLs); for the Botnet data set, we found 854 unique malicious attachments (in 3,008 emails with malicious attachments), and 900 unique malicious URLs (in 1,647 emails with malicious URLs); and for the User-Rep data set, we found 55,734 unique malicious attachments (in 102,239 emails with malicious attachments), and 1,508,276 unique malicious URLs (in 1,385,962 emails with malicious URLs); the higher number of unique URLs in this data set is due to many more spam emails that attempt to mimic emails of real companies with different URL paths for links in the email. If each unique attachment or URL represented one spam campaign (thus having similar features in campaign emails), then the diversity of these spam campaigns would be high, and this would strengthen our results because the classifiers can recognise a wide variety of spam campaigns with high reliability as measured by the AUC-PR and AUC-ROC scores for malicious attachments. In future work, we aim to address this issue by performing spam campaign analysis and to see if this will influence on classification results.

Overall, we partly confirm our hypothesis that emails with malicious attachments can be predicted from the features of the email text. Our evaluation on three real-world data sets composed only of spam emails shows the effects of data set size, the cumulative learning of potential spam emails over a year, and the importance of different features useful for classification. The work of identifying the more harmful types of spam email remains important if we are to prevent this vector for cybercrime by limiting exposure of malware to potential victims [Hofmeyr et al., 2013].

8.9 Summary

In this chapter, we presented rich descriptive sets of text features for the task of identifying spam emails with malicious attachments and URLs. We use three real-world

[7] http://www.trendmicro.com.au/cloud-content/us/pdfs/security-intelligence/white-papers/wp_blackhole-exploit-kit.pdf

data sets of spam emails, sourced from a manually labelled corpus (Habul), automated collection from spam traps (Botnet), and user reported spam emails from an email service provider (UserRep). Our results show that emails with malicious attachments can be reliably predicted using text features extracted only from emails, without requiring external resources. However, this is not the case with emails with malicious URLs as their text features do not differ much from emails with non-malicious URLs. We compared the classification performance for three classifiers: Naïve Bayes, Random Forest, and Support Vector Machine. We compared the selected features across our three data sets with the Random Forest classifier generally performing best. We have discussed the effects of differences in size of data set, the potential over-representation of spam campaign emails, and advantages and limitations of our approach. Our success suggested we might correctly identify over 95% of spam emails with malicious attachments without needing to scan the attachments. If this can be confirmed in subsequent research, a huge potential saving in resources used to detect and filter high-risk spam may be achieved. In addition, the methods will assist in the prevention of cybercrime, given that an estimated 6 billion emails with malicious attachments are sent every day.

In future work, we intend to add features to improve the classification of spam emails with malicious URLs. Indeed this form of delivering malware appears to be both evolving and now seems to be preferred to attachments containing malware. We aim to extract more features from the header of spam emails, such as graph relationships of common (anonymised) email addresses that could prove useful as alternative classifiers. One important unresolved issue is the possible effects of deliberate (and possibly repetitive) spam campaigns on classification results. We hope both to increase the size of scope of our real world data sets (adding for example prominent email data sets) and plan a comprehensive analysis combining and testing features, taking into account spam campaigns.

The next chapter concludes this thesis by summarising contributions and future work, and provides conclusions drawn from the research presented in this thesis.

Chapter 9

Conclusions and Future Work

In this thesis, we have developed a novel research area of cross-language vandalism detection (CLVD) and evaluated our techniques for five language editions (where possible) of Wikipedia: English (en), German (de), Spanish (es), French (fr), and Russian (ru). CLVD aims to develop language-independent machine learning models to address the focus of vandalism detection on the English Wikipedia, and the lack of policy and consistency in identifying vandalism cases in smaller Wikipedias of other languages. These CLVD machine learning models are building blocks to the development of automated vandalism detection algorithms on Wikipedia.

Past research in vandalism detection has focused almost entirely on the English Wikipedia, which we attributed to the language barrier being perceived as a limitation to study vandalism in other languages. Through our development of CLVD techniques and addressing its challenges, we have shown how to adapt state-of-the-art techniques to work across multiple languages, and how our proposed novel features and detection techniques improve detection rates and provide new directions for vandalism detection research.

The main challenges of CLVD that were addressed in Chapters 4 to 7 are (1) language independence, where features developed for machine learning algorithms must be applicable to multiple languages and have high detection rates of vandalism within these languages; and (2) extendibility, where classification models developed on these language independent features must be extendible to other languages by using different machine learning algorithms or finding suitable feature sets.

The additional challenges of vandalism detection addressed together with the CLVD challenges above are (3) high detection rate, which is difficult due to the ambiguity and constant changing nature of vandalism as new automatic counter-vandalism methods are introduced; (4) scalability, where automated detection algorithms must be able to process the large volume of Wikipedia to learn past vandalism and screen the influx of incoming edits ideally in real-time (when edits are submitted) during all hours, especially peak times; and (5) the variety of data, where the challenge is to transform the different raw metadata and text data provided by Wikipedia to features that allow machine learning algorithms to distinguish vandalism.

In this chapter, we summarise our contributions in each chapter of this thesis in Section 9.1 and outline our future directions of research in Section 9.2. Finally, we conclude this thesis in Section 9.3.

9.1 Summary of Contributions

We highlight below the main contributions of each chapter in this thesis for CLVD and vandalism detection in general. We focus on a summary and outcome as the aim and motivation were discussed in Section 1.1 on page 3.

- Chapter 2 provided the background to vandalism detection on Wikipedia, the need for CLVD through examples, current tools, past research contributions, research methodologies, evaluation measures, and experimental environment. In particular, this chapter described important data sets and data processing steps for the following research chapters of this thesis.
- Chapter 3 surveyed research on Wikipedia that covers multilingual aspects, where Wikipedia is used as a repository of multilingual knowledge for tasks such as machine translation, information sharing between languages through structured information boxes, large-scale analysis of differences between languages, and detecting plagiarism across languages. The chapter also covered vandalism aspects, such as characterisation, counter-vandalism tools, vandalism research data sets, and context-aware vandalism detection techniques. This chapter served as a summary of related research papers that motivated the research in the remaining chapters of this thesis.
- Chapter 4 proposed measures to summarise the vast information available in Wikipedia articles in different languages. We explored two important aspects of Wikipedia articles: the similarity of knowledge representation and coverage across languages, and the stability of articles over time. We showed how these measures allow identification of articles that have incomplete knowledge representations between languages, and the level of activity of editors. This chapter provided the data exploration of Wikipedia articles within and across languages as a step towards understanding vandalism in Wikipedia and what is needed across languages.
- Chapter 5 investigated vandalism detection using metadata with a particular focus on the suitability of different classification algorithms for CLVD. We demonstrated that the Gradient Tree Boosting (GTB) classification algorithm showed the best performance at detecting vandalism within and across languages at the expense of long training time, but the Random Forest (RF) classification algorithm is better overall with similar (but lower) classification scores than GTB with a significantly faster training time. We also demonstrated the novel use of the article views data set for vandalism detection, which may be showing changing view patterns of articles when vandalism happens. This chapter demonstrated that the RF algorithm is best overall, providing a balance in high classification scores and reasonable training time, and removed the need to compare different classifiers for CLVD in the following chapters.
- Chapter 6 continued the investigation of vandalism detection using text data for CLVD. This chapter proposed novel text features, compared and combined

these novel features with text features from past research, and demonstrated those novel features are more suitable for CLVD. In addition, we explored the contributions of bots (automated algorithms) – often ignored or dismissed in research – to detecting vandalism on Wikipedia. This chapter showed our contribution of novel text features suitable for CLVD and how bots (bot editors) and users (human editors) compare in the vandalism detection task across languages.

- Chapter 7 developed a novel context-aware CLVD technique to address types of sneaky vandalism that involve changing the meaning of text. The technique used word labels and sequential patterns of occurring words to identify words used for vandalism, which also allowed immediate identification of vandal words that constitute evidence of malicious intentions. We compared this technique with the text feature technique of the previous chapter, showing differences in the vandalism each technique detects. This chapter provided a new research direction for vandalism detection research of context-aware techniques to tackle increasingly difficult types of vandalism.
- Chapter 8 showed the extendibility of CLVD techniques to other domains by applying the text features from Chapter 6 to detecting malicious content in spam emails. Many of the novel features proposed in Chapter 6 and specific to this chapter showed that text in spam emails is a strong predictor of whether the attachments (and to a lesser extent, URLs) of emails are malicious. These text features and classification algorithms significantly reduce the need to scan emails using comparatively more complex data sources. This chapter demonstrated that the CLVD techniques in this thesis can be applied to other application domains, and provided a new direction of research to find some of the most damaging types of spam emails from a cybercrime perspective [Hofmeyr et al., 2013].

9.2 Future Work

In this section, we summarise the future research directions of this thesis based on the summary section of Chapters 4 to 8.

- **Additional languages.**[1] We aim to investigate new languages that are some of the largest language editions on Wikipedia[2]. In particular, the language editions of Vietnamese, Mandarin Chinese, and Japanese have become some of the largest Asian languages represented on Wikipedia in terms of the number of articles. The European languages used in this thesis share similar text structures which reduced the complexity of developing language independent features. The combination of Asian and European language families creates additional complex challenges because of the different representations of words,

[1] https://en.wikipedia.org/wiki/List_of_language_families
[2] https://meta.wikimedia.org/wiki/List_of_Wikipedias

sentences, and grammar that may cause difficulties in developing language independent features.

- **Cross-language summarisation measures.** The summarisation measures presented in Chapter 4 allowed visualisation and determination of the knowledge coverage (similarity) of articles across languages and activity (stability) of articles within languages. We intend to further refine these measures and develop new measures (such as incorporating semantic knowledge from DBpedia) to allow us to characterise the changes on Wikipedia in different languages.

- **Derived metadata features.** The metadata features presented in Chapter 5 were limited to existing features, which allowed fast combinations of feature data between two data sets. We look at deriving additional features based on monthly and yearly access patterns, and other more complex derived metadata features based on time, location, and reputation from West et al. [2010a].

- **Derived and additional text features.** Our proposed novel text features from Chapter 6 showed improved classification performance compared to text features used in related work. We intend to propose additional complex derived text features based on textual analysis and natural language processing to compare with the vast number of features used by West and Lee [2011].

- **Combined metadata and text features.** The CLVD research presented in Chapters 5 and 6 addressed the use of classifiers and contributions of bots and users, respectively. We look to combine these research aims in future work to comprehensively evaluate combinations of four distinguishing aspects of CLVD: classifiers, feature sets, user types, and languages.

- **Contributions of different user types.** Our experiments in Chapter 6 compared and contrasted the contributions of bots (bot editors) and users (human editors) because the counter-vandalism activities of bots are often not seen in research papers. Similarly, the differences between contributions of anonymous and registered users are often not distinguished in counter-vandalism research. These three user types have different contributions to detecting vandalism using metadata on the English Wikipedia as shown by West et al. [2010a]. We would like to extend our CLVD research to distinguish the three user types (bots, registered users, and anonymous users) and their contributions in future research.

- **Evaluation of other classifiers.** The Random Forest (RF) classifier was chosen for later chapters after experiments in Chapter 5 to avoid excessive results involving combinations of different classifiers, feature sets, and languages. We intend to revisit the experiments in Chapters 6 to 7 with different classifiers to explore whether there are more suitable classifiers that can meet the parallelism requirements and high classification scores of the RF classifier.

- **Additional semantic tags.** The tag set for the context-aware CLVD technique of Chapter 7 was limited to POS tags, but allowed us to demonstrate the feasibility and scalability of this novel technique for Wikipedia. We look to add new tag sets from different domains such as word semantics[3], WordNet[4], and Wikipedia ontologies[5]. More complex dependencies between these tag sets can also be modelled through feature functions to determine patterns of sneaky vandal words on Wikipedia.

- **Modelling additional dependencies.** The context-aware CLVD technique of Chapter 7 was limited to patterns of tags for sentences because of the linear-chain conditional random fields (CRF) classifier. The general CRF classifier [Sutton and McCallum, 2010] allows modelling additional dependencies between articles, which allows us to explore the spread of vandalism to adjacent internally linked articles, or articles linked across language.

- **Specific features for detecting malicious attachments and URLs.** In Chapter 8, we used text features from Chapter 6 and additional features for attachments and URLs. These additional features were relatively simple compared to the text features, which may not have allowed the classifier to distinguish the malicious content. We look to include additional features based on lexical analysis of names of attachments and URLs [Ma et al., 2009b; Le et al., 2011; Khami et al., 2014], and avoid using external resources where possible.

- **Spam campaign analysis.** In Section 8.8 of Chapter 8 on detecting malicious spam emails, one issue we did not address is spam campaigns in our email data sets because of non-disclosure of email sources (their originating servers) and the anonymisation of email addresses within each of our data sets. However, on review of the research for this thesis, our initial findings suggest that spam campaigns may be identifiable from approximate matching of text in different emails because although spam campaigns use templates [Stone-Gross et al., 2011], the variations for avoiding detection may not be significant enough to avoid text analysis for approximate matching [Kreibich et al., 2008].

- **Other collaborative environments.** Malicious activities such as vandalism also occur in other non-wiki based collaborative software systems. These systems face unique challenges such as preventing design flaws in experiments or tasks offered through the Amazon's Mechanical Turk[6], and preventing false map details or graffiti in the OpenStreetMap[7] [Neis et al., 2012]. We look to extend our research to these other collaborative systems to show the extendibility of our approaches in addition to the malicious spam email detection research of Chapter 8.

[3] https://www.freebase.com/
[4] http://wordnet.princeton.edu/
[5] http://dbpedia.org/About
[6] https://www.mturk.com/mturk/welcome
[7] http://www.openstreetmap.org

9.3 Conclusions

In this thesis, we have developed a novel research area of cross-language vandalism detection (CLVD) by presenting solutions to the problem of detecting vandalism across the languages of Wikipedia. Our primary contributions are a review of multilingual and vandalism detection research (Chapter 3), development of summary measures to understand information within and across the languages of Wikipedia (Chapter 4), demonstration of CLVD on metadata with a comparison of classifiers (Chapter 5), demonstration of CLVD on text data with an analysis of the contributions of bots (Chapter 6), development of a novel context-aware vandalism detection technique for sneaky types of vandalism that satisfies the challenges of CLVD (Chapter 7), and extension of the techniques of CLVD to another research domain of detecting malicious spam emails using the developed text features (Chapter 8). Overall, we hope our development of the novel CLVD research area provides new research directions for the vandalism detection communities on Wikipedia as well as other collaborative online environments to develop new generations and refine current generations of counter-vandalism bots.

Appendix A

Data Parsing and Processing

We expand in detail how we processed and parsed the vandalism cases from the Wikipedia data sets we downloaded.

A.1 Data Source

The Wikipedia database backup dumps are available for download at http://dumps.wikimedia.org/. The data dumps are available monthly for all Wikipedias. We downloaded the first available dump in January 2013 and ignored revisions dated in 2013 and after. Thus, the revisions we used are dated from early 2001 – when Wikipedia was publicly released – to the last second of December 2012. The data dumps we used are no longer available, but the current data dumps can be processed with the same timeline of revisions to obtain the same set of revisions.

The URLs to access our Wikipedia languages are:

- English: http://dumps.wikimedia.org/enwiki/ (2 January 2013, 64.0GB 7zip)
- German: http://dumps.wikimedia.org/dewiki/ (5 January 2013, 15.0GB 7zip)
- Spanish: http://dumps.wikimedia.org/eswiki/ (18 January 2013, 7.0GB 7zip)
- French: http://dumps.wikimedia.org/frwiki/ (4 January 2013, 11.0GB 7zip)
- Russian: http://dumps.wikimedia.org/ruwiki/ (15 January 2013, 6.6GB 7zip)

A.2 Data Processing

We present a detailed description of our data processing. We use the "Anarchism" article as a running example because it is one of the first articles created (and thus appear early in the data dump) and also frequently vandalised.

A.2.1 Revision Sample

The (full history) English Wikipedia data dump contains XML data about articles and their revisions. We show a sample of the XML below with some formatting and omissions of repeated content.

The article title and username are unique and thus used as identifiers on Wikipedia (along with the numerical IDs). We process each revision for the features shown in Table 3 of the paper. The <username> tag shows the name (or IP in <ip>) of the user that made the edit. We use a list of bot names to identify the revisions contributed by bots. In the next sections, we show how we analyse the comment to identify the repair of vandalism and take the difference of the revision text (between <text> and </text>) of the repaired revision and previous revision.

```
<page>
  <title>Anarchism</title>
  <ns>0</ns>
  <id>12</id>
  <revision>
    <id>233194</id>
    <timestamp>2001-10-11T20:18:47Z</timestamp>
    <contributor>
      <username>The Cunctator</username>
      <id>31</id>
    </contributor>
    <comment>*Restoring the deleted names until they get put somewhere else
        </comment>
    <text xml:space="preserve">''Anarchism'' is the political theory that
        advocates the abolition of all forms of government. The word
        anarchism derives from Greek roots &lt;i&gt;an&lt;/i&gt; (no) and &
        lt;i&gt;archos&lt;/i&gt; (ruler).

    ...
     </text>
    <sha1>supaosl5z6obobfp640do1905hdsrgt</sha1>
    <model>wikitext</model>
    <format>text/x-wiki</format>
  </revision>

  <revision>
    <id>233195</id>
    <parentid>233194</parentid>
    <timestamp>2001-11-28T13:32:25Z</timestamp>
    <contributor>
      <username>Ffaker</username>
      <id>157</id>
    </contributor>
    <comment>tolstoy and chomsky</comment>
    <text xml:space="preserve">''Anarchism'' is the political theory that
        advocates the abolition of all forms of government. The word
        anarchism derives from Greek roots &lt;i&gt;an&lt;/i&gt; (no) and &
        lt;i&gt;archos&lt;/i&gt; (ruler).

    ...
     </text>
```

```
    <sha1>ezk9gned3u61a0chomptja4wg6u5p34</sha1>
    <model>wikitext</model>
    <format>text/x-wiki</format>
  </revision>

  ...
</page>
```

A.2.2 Identifying Repaired Revisions

Each revision contains a comment (see `<comment>` in the previous section) from the user describing the change. In the case of repair of vandalism, we search the comment for two types of structure: free-form comments left by users and structured comments left by bots.

For bots and users, we search for the following patterns in the words of the comment, where some are language specific:

- rvv (revert due to vandalism)
- rv, rev, (note that "revert", "reverted", "reverting", and similar forms are matched by "rev")
- vandal, vandalism
- vandalismus (de)
- vandalismo (es)
- vandalisme (fr)
- вандал (ru), вандализм (ru)

There are some articles that were checked with more conditions, such as "Vandalism" titled articles and articles relating to the "Vandals" tribe of people[1]. These special articles are relatively much fewer than the rest of the other articles, so articles that were incorrectly labelled would not greatly affect the results.

Although bots have structured comments of the form "... Reverting ... vandalism by ... to version ...", we found that the above keywords were sufficient to identify their vandalism repairs.

A.2.3 Revision Diffs

Once we have identified the revision of the repair of vandalism, we take the diff of that revision and its previous revision, which gives us the edits made by the user. Below is a sample of the result vector of features summarising the repair of vandalism between revisions `11748407` and `11746850` by user `Kevehs`. This repair is seen on

[1] http://en.wikipedia.org/wiki/Vandals

Wikipedia at this URL: http://en.wikipedia.org/w/index.php?title=Anarchism&diff=11748407&oldid=11746850.

The diffs show the previous revision (assumed to contain vandalism) in the top section and the repaired revision in the bottom section. The three types of changes are + (added content), - (removed content), and ! (changed content). We show in bold the words showing vandalism, revision IDs, and the user making the edit.

```
Anarchism,12,11748407,2005-04-01T03:49:03Z,11746850,2005-04-01T03:15:24Z,20585,
Kevehs,0,rv vandalism stemming from ignorance and immaturity to last vs by pharos,
17766606d2a7c12ec0f1e26936d412ce76950eaa,1,
"*** 275,277 ****
!  'This page refers to anarchism as defined by pinko commie socialists.
For information on an alternative form of anarchism that actually makes "sense",
refer to:'  [[Anarcho-capitalism]]
-- 275,277 ---
!  'This page refers to anarchism as a philosophy of those who seek out and
identify structures of authority, hierarchy, and domination, and challenge them,
to increase the scope of human freedom.  This philosophy regards the social
hierarchies inherent in capitalism to be antithetical to freedom.  Other
philosophies, known as anti-state capitalism or anarcho-capitalism, take a
differing view, and sometimes use the &quot;anarchist&quot; label themselves
because they are opposed to governmental authority, though the tradition of the
anarchist movement rejects their use of this label.  For more information on anti-
state capitalism, refer to:'  [[Anarcho-capitalism]]"
```

A.2.4 Word Diffs

For the diff changes (marked by !), we take a sentence diff to identify the changed words. Below is a sample result vector of the previous section. We calculated some non-word features at this point, and then calculated the word features presented in Chapters 6 and 7. Note that we have built and tested more features than presented in those chapters, but we filtered them for training and testing to obtain the presented results.

```
Anarchism,2005-04-01T03:49:03Z,2005-04-01T03:15:24Z,2019,0,20585,
Kevehs,0,rv vandalism stemming from ignorance and immaturity to last vs by pharos,
0,0,1,1,0,0,191,669,-268,0.33827160493827163,-72,0.28,
"'anarchism': 1, 'form': 1, 'pinko': 1, 'that': 1,'defined': 1, 'an': 1,
'socialists': 1, 'commie': 1,'sense': 1, 'alternative': 1, 'actually': 1,
'makes': 1, 'by': 1",13,13
```

Bibliography

Adafre, S. F. and de Rijke, M., 2006. Finding Similar Sentences across Multiple Languages in Wikipedia. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. (cited on page 29)

Adar, E.; Skinner, M.; and Weld, D. S., 2009. Information Arbitrage Across Multilingual Wikipedia. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining (WSDM)*. (cited on page 29)

Adler, B. T. and de Alfaro, L., 2007. A Content-Driven Reputation System for the Wikipedia. In *Proceedings of the 16th International World Wide Web Conference (WWW)*. (cited on pages 35 and 37)

Adler, B. T.; de Alfaro, L.; Mola-velasco, S. M.; Rosso, P.; and West, A. G., 2011. Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*. (cited on pages 35, 80, 89, 98, 106, 108, and 111)

Adler, B. T.; de Alfaro, L.; and Pye, I., 2010. Detecting Wikipedia Vandalism using WikiTrust - Lab Report for PAN at CLEF 2010. In *CLEF (Notebook Papers/Labs/Workshops)*. (cited on pages 61, 89, and 111)

Adler, B. T.; de Alfaro, L.; Pye, I.; and Raman, V., 2008. Measuring Author Contributions to the Wikipedia. In *Proceedings of the 4th International Symposium on Wikis (WikiSym)*. (cited on pages 73, 81, and 88)

Alazab, M. and Broadhurst, R., 2014. Spam and Criminal Activity. In *Trends and Issues (Australian Institute of Criminology)*. (cited on page 121)

Altman, N. S., 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46 (1992), 175–185. (cited on page 65)

Ameel, E.; Storms, G.; Malt, B. C.; and Assche, F. V., 2009. Semantic Convergence in the Bilingual Lexicon. *Journal of Memory and Language*, 60 (2009), 270–290. (cited on page 40)

Bao, P.; Hecht, B.; Carton, S.; Quaderi, M.; Horn, M.; and Gergle, D., 2012. Omnipedia: Bridging the Wikipedia Language Gap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. (cited on page 30)

Basaras, P.; Katsaros, D.; and Tassiulas, L., 2013. Detecting Influential Spreaders in Complex, Dynamic Networks. *Computer*, 46 (2013), 24–29. (cited on page 117)

Bentley, J. L., 1975. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 18 (1975), 509–517. (cited on page 65)

Blanzieri, E. and Bryl, A., 2008. A Survey of Learning-Based Techniques of Email Spam Filtering. *Artificial Intelligence Review*, 29 (2008), 63–92. (cited on pages 115, 116, 117, 119, and 129)

Boroditsky, L., 2011. How Language Shapes Thought. *Scientific American*, 304 (2011), 63–65. (cited on pages 40 and 46)

Bottou, L. and Bousquet, O., 2008. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems*. NIPS Foundation. (cited on page 65)

Bouma, G.; Duarte, S.; and Islam, Z., 2009. Cross-lingual Alignment and Completion of Wikipedia Templates. In *Proceedings of the 3rd International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3)*. (cited on page 29)

Breiman, L., 2001. Random Forests. *Machine learning*, 45 (2001), 5–32. (cited on page 65)

Breiman, L.; Friedman, J.; Stone, C. J.; and Olshen, R. A., 1984. *Classification and Regression Trees*. CRC press. (cited on page 64)

Broadhurst, R.; Grabosky, P.; Alazab, M.; Bouhours, B.; Chon, S.; and Da, C., 2013. Crime in Cyberspace: Offenders and the Role of Organized Crime Groups. In *Social Science Research Network (SSRN)*. (cited on page 120)

Callison-Burch, C.; Koehn, P.; Monz, C.; and Zaidan, O. F., 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*. (cited on page 44)

Canali, D.; Cova, M.; Vigna, G.; and Kruegel, C., 2011. Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages. In *Proceedings of the 20th International World Wide Web Conference (WWW)*. (cited on page 118)

Chin, S.-C. and Street, W. N., 2012. Divide and Transfer: an Exploration of Segmented Transfer to Detect Wikipedia Vandalism. *Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings*, 27 (2012), 133–144. (cited on pages 32, 89, and 111)

Chin, S.-C.; Street, W. N.; Srinivasan, P.; and Eichmann, D., 2010. Detecting Wikipedia Vandalism with Active Learning and Statistical Language Models. In *Proceedings of the 4th Workshop on Information Credibility (WICOW)*. (cited on pages 35 and 61)

Christen, P., 2012. *Data Matching*. Springer Berlin Heidelberg. (cited on pages 29, 40, and 46)

Cilibrasi, R. L. and Vitanyi, P. M., 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19 (2007), 370–383. (cited on page 38)

Costa-Jussa, M. R. and Farrus, M., 2014. Statistical Machine Translation Enhancements Through Linguistic Levels: A Survey. *ACM Computing Surveys*, 46 (2014), 42:1–42:28. (cited on pages 27 and 28)

Cover, T. M. and Thomas, J. A., 2012. *Elements of Information Theory*. John Wiley & Sons. (cited on page 49)

Davis, J. and Goadrich, M., 2006. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine learning (ICML)*. (cited on pages 24, 25, and 86)

Eshete, B. and Villafiorita, A., 2013. Effective Analysis, Characterization, and Detection of Malicious Web Pages. In *Proceedings of the 22nd International Conference on World Wide Web Companion*. (cited on page 118)

Federico, M.; Bertoldi, N.; and Cettolo, M., 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. (cited on page 42)

Felegyhazi, M.; Kreibich, C.; and Paxson, V., 2010. On the Potential of Proactive Domain Blacklisting. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET)*, 6–6. USENIX Association. (cited on page 118)

Filatova, E., 2009. Multilingual Wikipedia, Summarization, and Information Trustworthiness. In *Proceedings of the SIGIR Workshop on Information Access in a Multilingual World*. (cited on pages 29, 30, and 54)

Friedman, J. H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, (2001), 1189–1232. (cited on page 65)

Gabrilovich, E. and Markovitch, S., 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. (cited on page 46)

Geiger, R. S., 2011. *The Lives of Bots*. Institute of Network Cultures, Amsterdam. (cited on pages 2, 34, 88, 91, and 93)

Geiger, R. S. and Halfaker, A., 2013. When the Levee Breaks: Without Bots, What Happens to Wikipedia's Quality Control Processes? In *Proceedings of the 9th International Symposium on Open Collaboration (WikiSym)*. (cited on pages 9, 33, and 34)

Geiger, R. S. and Ribes, D., 2010. The Work of Sustaining Order in Wikipedia: The Banning of a Vandal. In *Proceedings of the 22nd ACM Conference on Computer Supported Cooperative Work (CSCW)*. (cited on pages 33, 73, and 81)

Gipp, B., 2014. *Plagiarism Detection*. Springer Fachmedien Wiesbaden. (cited on page 31)

Gomaa, W. H. and Fahmy, A. A., 2013. A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, (2013). (cited on page 29)

González-Brenes, J. P.; Mostow, J.; and Duan, W., 2011. How to Classify Tutorial Dialogue? Comparing Feature Vectors vs. Sequences. In *Proceedings of the 4th International Conference on Educational Data Mining (EDM)*. (cited on page 106)

Halfaker, A.; Geiger, R. S.; and Terveen, L., 2014. Snuggle: Designing for efficient socialization and ideological critique. In *Proceedings of the 2014 SIGCHI Conference on Human Factors in Computing Systems (CHI)*. (cited on pages 8, 14, 16, 33, and 35)

Halfaker, A.; Kittur, A.; and Riedl, J., 2011. Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration (WikiSym)*. (cited on pages 8, 24, 33, and 88)

Halfaker, A. and Riedl, J., 2012. Bots and Cyborgs: Wikipedia's Immune System. *Computer*, 45 (2012), 79–82. (cited on pages 34, 73, 88, and 91)

Hao, S.; Syed, N. A.; Feamster, N.; Gray, A. G.; and Krasser, S., 2009. Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine. In *Proceedings of the 18th Conference on USENIX Security Symposium (SSYM)*, vol. 9. (cited on page 117)

Harpalani, M.; Hart, M.; Singh, S.; Johnson, R.; and Choi, Y., 2011. Language of Vandalism: Improving Wikipedia Vandalism Detection via Stylometric Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT): Short Papers*. (cited on pages 36, 89, and 111)

Hasan, H. and Pfaff, C. C., 2006. The Wiki: an environment to revolutionise employees' interaction with corporate knowledge. In *Proceedings of the 18th Australia conference on Computer-Human Interaction (OZCHI)*, 377–380. ACM. (cited on page 1)

Hecht, B. and Gergle, D., 2010. The Tower of Babel Meets Web 2.0: User-Generated Content and Its Applications in a Multilingual Context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. (cited on page 30)

Hofmeyr, S.; Moore, T.; Forrest, S.; Edwards, B.; and Stelle, G., 2013. Modeling Internet-Scale Policies for Cleaning up Malware. In *Economics of Information Security and Privacy III*, 149–170. Springer. (cited on pages 117, 134, and 139)

Hutchins, W. J. and Somers, H. L., 1992. *An Introduction to Machine Translation*. Academic Press London. (cited on page 57)

Itakura, K. Y. and Clarke, C. L., 2009. Using Dynamic Markov Compression to Detect Vandalism in the Wikipedia. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. (cited on page 37)

Javanmardi, S.; McDonald, D. W.; and Lopes, C. V., 2011. Vandalism Detection in Wikipedia: A High-Performing, Feature-Rich Model and its Reduction Through Lasso. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration (WikiSym)*. (cited on pages 35, 89, 93, 106, 108, and 111)

John, J. P.; Moshchuk, A.; Gribble, S. D.; and Krishnamurthy, A., 2009. Studying Spamming Botnets Using Botlab. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. (cited on page 120)

Khami, A.; Baharudin, B.; and Jung, L., 2014. Characterizing A Malicious Web Page. *Australian Journal of Basic and Applied Sciences*, 8, 3 (2014), 69–76. (cited on pages 118 and 141)

Kittur, A.; Chi, E. H.; and Suh, B., 2008. Crowdsourcing User Studies With Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 453–456. ACM. (cited on pages 1 and 8)

Kittur, A.; Suh, B.; Pendleton, B. A.; and Chi, E. H., 2007. He Says, She Says: Conflict and Coordination in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. (cited on pages 1, 2, 9, 19, 32, 33, 61, and 93)

Koehn, P., 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit*. (cited on page 42)

Koehn, P., 2009. *Statistical Machine Translation*. Cambridge University Press. (cited on page 28)

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E., 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (ACL)*. (cited on pages 28, 40, and 42)

Kothari, C. R., 2004. *Research Methodology: Methods and Techniques*. New Age International Publishers. (cited on page 21)

Kreibich, C.; Kanich, C.; Levchenko, K.; Enright, B.; Voelker, G. M.; Paxson, V.; and Savage, S., 2008. On the Spam Campaign Trail. In *Proceedings of the 1st USENIX Conference on Large-scale Exploits and Emergent Threats (LEET)*. (cited on page 141)

Kreibich, C.; Kanich, C.; Levchenko, K.; Enright, B.; Voelker, G. M.; Paxson, V.; and Savage, S., 2009. Spamcraft: An Inside Look At Spam Campaign Orchestration. In *Proceedings of the 2nd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET)*. (cited on page 125)

Kudo, T., 2013. CRF++: Yet Another CRF toolkit. (cited on pages 94 and 100)

Kulkarni, R. G.; Trivedi, G.; Suresh, T.; Wen, M.; Zheng, Z.; and Rose, C., 2012. Supporting Collaboration in Wikipedia Between Language Communities. In *Proceedings of the 4th International Conference on Intercultural Collaboration (ICIC)*. (cited on page 29)

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*. (cited on pages 94, 98, and 99)

Laurent, M. R. and Vickers, T. J., 2009. Seeking Health Information Online: Does Wikipedia Matter? *Journal of the American Medical Informatics Association (JAMIA)*, 16 (2009), 471–479. (cited on page 63)

Le, A.; Markopoulou, A.; and Faloutsos, M., 2011. PhishDef: URL Names Say It All. In *IEEE INFOCOM*. (cited on pages 116, 118, and 141)

Lee, M., 2012. Who's next? Identifying Risks Factors for Subjects of Targeted Attacks. In *Proceedings of the Virus Bulletin Conference*. (cited on page 116)

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; and Bizer, C., 2014. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, (2014). (cited on page 30)

Likarish, P.; Jung, E. E.; and Jo, I., 2009. Obfuscated Malicious Javascript Detection using Classification Techniques. In *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE)*. (cited on pages 116 and 118)

Liu, Y.; Dai, L.; Zhou, W.; and Huang, H., 2012. Active Learning for Cross Language Text Categorization. In *Proceedings of the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*. Springer Berlin Heidelberg. (cited on page 66)

Ma, J.; Saul, L. K.; Savage, S.; and Voelker, G. M., 2009a. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (cited on pages 116 and 118)

Ma, J.; Saul, L. K.; Savage, S.; and Voelker, G. M., 2009b. Identifying Suspicious URLs: An Application of Large-Scale Online Learning. In *Proceedings of the 26th*

Annual International Conference on Machine Learning (ICML). (cited on pages 118 and 141)

Ma, J.; Saul, L. K.; Savage, S.; and Voelker, G. M., 2011. Learning to Detect Malicious URLs. In *ACM Transactions on Intelligent Systems and Technology (TIST)*. (cited on pages 116 and 118)

Madnani, N. and Dorr, B. J., 2010. Generating Phrasal and Sentential Paraphrases: A Survey of Data-driven Methods. *Computational Linguistics*, 36, 3 (2010), 341–387. (cited on page 46)

Manning, C. D.; Raghavan, P.; and Schütze, H., 2008. *Introduction to Information Retrieval*. Cambridge University Press. (cited on pages 30, 40, 46, and 54)

Martin, S.; Sewani, A.; Nelson, B.; Chen, K.; and Joseph, A. D., 2005. Analyzing Behaviorial Features for Email Classification. In *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS)*. (cited on page 118)

Martinez, A. R., 2012. Part-of-speech tagging. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4 (2012), 107–113. (cited on pages 96 and 112)

Massey, F. J., 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46 (1951), 68–78. (cited on pages 75 and 78)

Masud, M. M.; Khan, L.; and Thuraisingham, B., 2007. Feature Based Techniques for Auto-Detection of Novel Email Worms. In *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*. (cited on page 118)

Medelyan, O.; Milne, D.; Legg, C.; and Witten, I. H., 2009. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67 (2009), 716–754. (cited on page 46)

Mohammadi, M. and Ghasem-Aghaee, N., 2010. Building Bilingual Parallel Corpora Based on Wikipedia. In *Proceedings of the 2nd International Conference on Computer Engineering and Applications (ICCEA)*. (cited on pages 29 and 46)

Mola-Velasco, S. M., 2010. Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals. In *CLEF (Notebook Papers/Labs/-Workshops)*. (cited on pages 35, 75, 77, 78, 89, 106, 108, 111, 119, 123, 125, and 126)

Neis, P.; Goetz, M.; and Zipf, A., 2012. Towards automatic vandalism detection in OpenStreetMap. *International Journal of Geo-Information (IJGI)*, 1, 3 (2012), 315–332. (cited on pages 1, 8, and 141)

Oliver, J.; Cheng, S.; Manly, L.; Zhu, J.; Paz, R. D.; Sioting, S.; and Leopando, J., 2012. Blackhole Exploit Kit: A Spam Campaign, Not a Series of Individual Spam Runs. In *Trend Micro Incorporated*. (cited on page 134)

Pan, S. J. and Yang, Q., 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22 (2010), 1345–1359. (cited on pages 22 and 32)

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J., 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*. (cited on page 44)

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Édouard Duchesnay, 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research (JMLR)*, 12 (2011), 2825–2830. (cited on pages 64, 78, 80, 108, 127, and 129)

Pereira, R. C., 2010. *Cross-Language Plagiarism Detection*. Master's thesis, Universidade Federal do Rio Grande do Sul. (cited on page 31)

Potthast, M., 2010. Crowdsourcing a Wikipedia Vandalism Corpus. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. (cited on pages 1, 3, 4, 9, 17, 18, 32, 61, 74, and 81)

Potthast, M.; Barrón-Cedeño, A.; Stein, B.; and Rosso, P., 2011. Cross-Language Plagiarism Detection. *Language Resources and Evaluation*, 45 (2011), 45–62. (cited on pages 31 and 68)

Potthast, M. and Gerling, R., 2007. Wikipedia Vandalism Corpus Webis-WVC-07. http://www.uni-weimar.de/medien/webis/research/corpora. (cited on page 32)

Potthast, M.; Stein, B.; and Gerling, R., 2008. Automatic Vandalism Detection in Wikipedia. In *Proceedings of the 30th European Conference on IR Research (ECIR): Posters*. (cited on page 37)

Potthast, M.; Stein, B.; and Holfeld, T., 2010. Overview of the 1st International Competition on Wikipedia Vandalism Detection. In *CLEF (Notebook Papers/Labs/-Workshops)*. (cited on page 68)

Priedhorsky, R.; Chen, J.; Lam, S. K.; Panciera, K.; Terveen, L.; and Riedl, J., 2007. Creating, Destroying, and Restoring Value in Wikipedia. In *Proceedings of the ACM International Conference on Supporting Group Work (GROUP)*. (cited on pages 1, 9, 10, 33, 61, and 88)

Quanz, B.; Huan, J.; and Mishra, M., 2012. Knowledge Transfer with Low-Quality Data: a Feature Extraction Issue. In *Proceedings of the 27th International Conference on Data Engineering (ICDE)*. (cited on page 22)

Quinlan, J. R., 1993. *C4.5: Programs for Machine Learning*, vol. 1. Morgan Kaufmann. (cited on page 64)

Ramachandran, A.; Dagon, D.; and Feamster, N., 2006. Can DNS-Based Blacklists Keep Up with Bots? In *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS)*. (cited on page 118)

Ramaswamy, L.; Tummalapenta, R. S.; Li, K.; and Pu, C., 2013. A Content-Context-Centric Approach for Detecting Vandalism in Wikipedia. In *Proceedings of the 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*. (cited on pages 37, 94, 98, 110, and 112)

Rehurek, R. and Sojka, P., 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC 2010 Workshop New Challenges for NLP Frameworks*. (cited on page 46)

Rigutini, L.; Maggini, M.; and Liu, B., 2005. An EM Based Training Algorithm for Cross-Language Text Categorization. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. (cited on page 66)

Rzeszotarski, J. and Kittur, A., 2012. Learning from History: Predicting Reverted Work at the Word Level in Wikipedia. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. (cited on page 37)

Schmid, H., 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP)*. (cited on pages 94, 96, and 97)

Schmid, H., 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*. (cited on page 97)

Sen, S.; Li, T. J.-J.; Lesicko, M.; Weiland, A.; Gold, R.; Li, Y.; Hillmann, B.; and Hecht, B., 2014. WikiBrain: Democratizing computation on Wikipedia. In *Proceedings of the 10th International Symposium on Open Collaboration (OpenSym)*. (cited on pages 8 and 30)

Smets, K.; Goethals, B.; and Verdonk, B., 2008. Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach. In *Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence (WikiAI)*. (cited on pages 36 and 37)

Smith, J. R.; Quirk, C.; and Toutanova, K., 2010. Extracting Parallel Sentences from Comparable Corpora using Document Level Alignment. In *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*. (cited on page 29)

Steiner, T., 2014. Bots vs. Wikipedians, Anons vs. Logged-Ins. In *Proc. the 23rd Int'l World Wide Web Conference Companion - Poster - Web Science Track*. (cited on page 34)

Stone-Gross, B.; Holz, T.; Stringhini, G.; and Vigna, G., 2011. The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns. In *Proceedings of the 4th USENIX conference on Large-scale Exploits and Emergent Threats (LEET)*. (cited on pages 120, 125, and 141)

Straight, R. J., 2014. Whatever You're Doing Isn't Good Enough: Paradigm Shift in Approach to Cybersecurity Needed to Minimize Exposure, Liability and Loss. In *FinTech Law Report*. (cited on page 116)

Suh, B.; Convertino, G.; Chi, E. H.; and Pirolli, P., 2009. The Singularity is Not Near: Slowing Growth of Wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration (WikiSym)*. (cited on pages 7, 39, and 40)

Sumbana, M.; Gonçalves, M. A.; Silva, R.; Almeida, J.; and Veloso, A., 2012. Automatic Vandalism Detection in Wikipedia with Active Associative Classification. In *Proceedings of the 2nd International Conference on Theory and Practice of Digital Libraries (TPDL)*. Springer Berlin Heidelberg. (cited on page 36)

Sutton, C. and McCallum, A., 2010. An Introduction to Conditional Random Fields. In *ARXIV*. (cited on pages 98, 99, 103, 112, and 141)

Thomas, C. and Sheth, A. P., 2007. Semantic Convergence of Wikipedia Articles. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. (cited on pages 30, 46, 47, 48, and 54)

Tran, K.-N.; Alazab, M.; and Broadhurst, R., 2013. Towards a Feature Rich Model for Predicting Spam Emails containing Malicious Attachments and URLs. In *Proceedings of the 11th Australasian Data Mining Conference (AusDM)*. (cited on page 5)

Tran, K.-N. and Christen, P., 2013a. Cross-Language Prediction of Vandalism on Wikipedia Using Article Views and Revisions. In *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. (cited on pages 5, 59, and 115)

Tran, K.-N. and Christen, P., 2013b. Identifying Multilingual Wikipedia Articles based on Cross Language Similarity and Activity. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM) - Poster Track*. (cited on pages 4 and 39)

Tran, K.-N. and Christen, P., 2014. Cross-Language Learning from Bots and Users to detect Vandalism on Wikipedia. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, (2014). (cited on pages 5 and 73)

Viegas, F. B.; Wattenberg, M.; and Dave, K., 2004. Studying Cooperation and Conflict between Authors with history flow Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. (cited on pages 9, 32, and 33)

Wang, W. Y. and McKeown, K. R., 2010. "Got You!": Automatic Vandalism Detection in Wikipedia with Web-based Shallow Syntactic-Semantic Modeling. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling)*. (cited on pages 35, 98, 102, and 103)

Wang, X.; Yu, W.; Champion, A.; Fu, X.; and Xuan, D., 2007. Detecting Worms via Mining Dynamic Program Execution. In *Proceedings of the International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm)*. (cited on page 118)

Wasikowski, M. and Chen, X.-W., 2010. Combating the Small Sample Class Imbalance Problem Using Feature Selection. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22 (2010), 1388–1400. (cited on page 22)

West, A. G., 2013. *Damage Detection and Mitigation in Open Collaboration Applications*. Ph.D. thesis, University of Pennsylvania. (cited on pages 1 and 60)

West, A. G.; Kannan, S.; and Lee, I., 2010a. Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata. In *Proceedings of the 3rd European Workshop on System Security (EUROSEC)*. (cited on pages 36, 93, 98, and 140)

West, A. G.; Kannan, S.; and Lee, I., 2010b. STiki: An Anti-vandalism Tool for Wikipedia Using Spatio-temporal Analysis of Revision Metadata. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration (WikiSym)* (Gdansk, Poland, 2010). (cited on pages 14, 16, and 33)

West, A. G. and Lee, I., 2011. Multilingual Vandalism Detection using Language-Independent & Ex Post Facto Evidence. In *CLEF (Notebook Papers/Labs/Workshops)*. (cited on pages 3, 32, 36, 75, 77, 78, 88, 89, 93, 106, 108, 111, 119, 123, 125, and 140)

White, J. and Maessen, R., 2010. ZOT! to Wikipedia Vandalism. In *CLEF (Notebook Papers/Labs/Workshops)*. (cited on page 68)

Wu, Q.; Irani, D.; Pu, C.; and Ramaswamy, L., 2010. Elusive Vandalism Detection in Wikipedia: A Text Stability-based Approach. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*. (cited on pages 35, 37, 61, 110, and 112)

Yeung, C. A.; Duh, K.; and Nagata, M., 2011. Providing Cross-Lingual Editing Assistance to Wikipedia Editors. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*. (cited on pages 29, 30, 40, 46, and 54)

Yuncong, C. and Fung, P., 2010. Unsupervised Synthesis of Multilingual Wikipedia Articles. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling)*. (cited on page 46)