Texas A&M University - Commerce

Department of Computer Science

Machine Learning for Disaster Detection through Twitter Analysis

Sneha Perithambi

Supervisor: Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of
Texas A&M University - Commerce for the degree of
Master of Science in *Computer Science*

February 5, 2024

Declaration

I, Firstname(s) Lastname, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Sneha Perithambi
February 5, 2024

Abstract

Twitter has become a crucial medium for people to use their smartphones to provide real-time views of events in the context of modern disaster communication. The difficulty, therefore, lies in programmatically separating the language used in tweets to convey metaphors from actual news of calamities. In order to determine if a tweet is indeed connected to a crisis, this study presents a machine learning algorithm. The suggested approach uses a painstakingly hand-classified dataset of 10,000 tweets and combines vectorization, classification, and NLP models to improve prediction accuracy. By tackling the challenges presented by metaphorical language, this research helps to construct a sophisticated machine learning framework that can determine the genuine nature of tweets connected to emergencies.

Keywords: twitter, real time, text analysis, NLP, disaster detection

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences

Chapter 1

Introduction

Twitter has emerged as a ubiquitous platform for event reporting, facilitated by the widespread use of smartphones. This dynamic environment offers unparalleled opportunities for immediate and decentralized communication, especially during critical events such as disasters. The advent of this digital era has underscored the pressing need for effective crisis communication strategies to harness the potential of Twitter as a valuable tool for situational awareness and emergency response.

However, amidst the wealth of real-time data flowing through Twitter feeds, a significant challenge arises in the accurate identification and differentiation of metaphorical expressions from authentic crisis-related information within tweets. Metaphors, while a powerful linguistic tool for expression, often introduce ambiguity and complexity, posing a considerable hurdle in the quest for reliable crisis detection. The inherent nature of metaphorical language requires a nuanced understanding that transcends traditional analytical approaches, demanding innovative solutions to decipher the true intent behind tweets during critical events.

This research embarks on the journey to address this multifaceted challenge by delving into the intricacies of Twitter communication during crises. The aim is to develop a sophisticated machine learning framework capable of distinguishing between metaphorical language and genuine crisis-related information. Leveraging the prevalence of smartphones and the instantaneous nature of Twitter reporting, this study seeks to contribute to the advancement of crisis communication strategies, fostering a more effective and accurate response to emergencies in the digital age.

1.1 Background

The motivation stems from the critical need for effective crisis communication strategies in utilizing Twitter as a valuable tool for situational awareness and emergency response.

The central challenge lies in accurately discerning metaphorical expressions from genuine crisis-related information within tweets. Metaphors, while powerful for expression, introduce ambiguity, posing a significant hurdle to reliable crisis detection. This project addresses this challenge through the development of a sophisticated machine learning framework, drawing on established classification algorithms with intentional omission of specific names for flexibility. Hyperparameter tuning and model selection are explored for optimization. Concurrently, Natural Language Processing (NLP) models capture contextual nuances to enhance metaphorical language understanding.

1.2 Problem statement

The significant challenge in the realm of disaster monitoring involves distinguishing disaster-related tweets from general Twitter content. This study aims to develop a machine learning algorithm capable of addressing this challenge and accurately determining if a tweet is genuinely connected to a crisis.

1.3 Aims and objectives

Aims: The primary aim of this project is to enhance the field of disaster monitoring on Twitter by developing a sophisticated machine learning framework. The goal is to accurately distinguish tweets related to disasters from the broader spectrum of general content on the platform. Through this endeavor, we seek to contribute to the improvement of crisis communication strategies in the digital age.

Objectives: Implement data exploration and preprocessing techniques to ensure the dataset is prepared for training and evaluation. Explore and apply various machine learning classification algorithms for the effective categorization of tweets, emphasizing the optimization of hyperparameters and model selection.

Incorporate Natural Language Processing (NLP) models to capture contextual nuances, enhancing the understanding of metaphorical language within tweets. Evaluate the performance of the developed framework using rigorous metrics to ensure accuracy and reliability in distinguishing disaster-related tweets. Provide insights and recommendations for advancing crisis communication strategies based on the project outcomes.

1.4 Solution approach

The solution approach consists of distinct stages, including data collection, exploration, preprocessing, and the implementation of algorithms for classification and NLP.

1.4.1 Data Collection, Exploration, and Preprocessing

Data Exploration: Comprehensive exploration is conducted to understand the characteristics of the dataset, including the distribution of metaphorical expressions and crisis-related content. **Preprocessing:** Textual data undergoes preprocessing, including tokenization, stemming, and handling of special characters, to prepare it for subsequent stages.

1.4.2 Implementation of Algorithms

Established machine learning classification algorithms are implemented to effectively classify tweets. Hyperparameter tuning and model selection are explored to optimize performance. NLP models are implemented to capture contextual nuances and improve the understanding of metaphorical language in tweets.

1.5 Summary of contributions and achievements

This research contributes a sophisticated machine learning framework capable of distinguishing metaphorical language from genuine crisis-related information on Twitter. Achievements include the development of a robust algorithm, leveraging a hand-classified dataset for effective model training.

1.6 Organization of the report

The report follows a structured format, exploring background, problem statement, solution approach, detailed methodologies, results, discussions, and conclusions.

Chapter 2

Literature Review

Recognizing the imperative for automated solutions to distinguish genuine disaster-related tweets from metaphorical or unrelated content, the research gets crucial reference, aiding in exploring effective approaches and refining the accuracy of disaster-related tweet prediction models from this Kaggle competition Addison Howard (2019) Phil Culliton (2019). With widespread smartphone use, Twitter becomes a primary source for disaster-related information. Utilizing a dataset of 10,000 hand-classified tweets, the goal is to employ machine learning models for binary classification and NLP to predict tweets genuinely related to disasters.

2.1 Related Work

In the research Chanda (2021) extensive evaluation of Deep Learning methods for classifying disaster-related tweets. Identification of preprocessing steps, emphasizing named entity substitution.Performance analysis of custom neural networks and Transformer models.Emphasis on practical application potential for automatic disaster detection. BERT is used in Deb and Chanda (2022) where exploration of Twitter's real-time data for disaster identification and challenges in manual data processing due to volume is elaborated. Evaluation of BERT embeddings' superiority in disaster prediction, compared to traditional word embeddings. Discussion on opportunities and challenges of BERT embeddings in Twitter sentiment analysis. Detailed analysis for various algorithms are explored in Fontalis et al. (2023) Theoretical basis of various ML algorithms for tweet analysis (BNB, MNB, LR, KNN, DT, RF).Process flow from dataset import to model training, emphasizing the importance of Exploratory Data Analysis (EDA). Selection of ML models based on suitability, using Wordclouds for identifying relevant words.Explanation of Bayesian algorithms, logistic regression, decision tree, and random forest usage.Iparraguirre-Villanueva et al. (2023)Exploration of BERT embeddings' effectiveness in disaster prediction on social media. Overview of challenges in manual disaster identification due to data volume. Application of different word embeddings (BOW, context-free, contextual) in disaster prediction models. Utilization of embeddings in both traditional ML methods and neural network-based models.In Saddam et al. (2023)Studies introduce normalization processes for words with similar meanings. Stemming, using the Indonesian literary library in Python, maximizes text processing efficieny. Machine learning involves labeling real opinions, crucial for SVM model training. SVM models, especially for multiclass classification, are discussed for sentiment analysis. K-Fold Cross Validation ensures robust testing, evaluating accuracy, precision, recall, and F-score. Confusion matrices aid in a

comprehensive understanding of model performance.

2.2 Critique of the review

Existing research demonstrates the effectiveness of advanced techniques such as deep learning models and contextual embeddings for disaster-related sentiment analysis. Critique emphasizes the need for continuous improvement in addressing challenges related to bias mitigation and contextual understanding.

2.3 Summary

The literature review underscores the significance of sophisticated preprocessing techniques, advanced text processing, and the application of machine learning models for disaster-related sentiment analysis on Twitter. Key findings highlight the potential of deep learning and contextual embeddings in achieving accurate disaster detection, with practical implications for real-world applications. Continuous improvement is encouraged to address existing challenges and further enhance the reliability of sentiment analysis systems.

Chapter 3

Methodology

We mentioned in Chapter 1 that a project report's structure could follow a particular paradigm. Hence, the organization of a report (effectively the Table of Content of a report) can vary depending on the type of project you are doing. Check which of the given examples suit your project. Alternatively, follow your supervisor's advice.

3.1 Examples of the sections of a methodology chapter

A general report structure is summarised (suggested) in Table 3.1. Table 3.1 describes that, in general, a typical report structure has three main parts: (1) front matter, (2) main text, and (3) end matter. The structure of the front matter and end matter will remain the same for all the undergraduate final year project report. However, the main text varies as per the project's needs.

3.1.1 Example of a software/Web development main text structure

Notice that the "methodology" Chapter of Software/Web development in Table 3.2 takes a standard software engineering paradigm (approach). Alternatively, these suggested sections can be the chapters of their own. Also, notice that "Chapter 5" in Table 3.2 is "Testing and Validation" which is different from the general report template mentioned in Table 3.1. Check with your supervisor if in doubt.

3.1.2 Example of an algorithm analysis main text structure

Some project might involve the implementation of a state-of-the-art algorithm and its performance analysis and comparison with other algorithms. In that case, the suggestion in Table 3.3 may suit you the best.

3.1.3 Example of an application type main text structure

If you are applying some algorithms/tools/technologies on some problems/datasets/etc., you may use the methodology section prescribed in Table 3.4.

Table 3.1: Undergraduate report template structure

Frontmatter		Title Page Abstract Acknowledgements Table of Contents List of Figures List of Tables List of Abbreviations
Main text	Chapter 1	Introduction
	Chapter 2	Literature Review
	Chapter 3	Methodology
	Chapter 4	Results
	Chapter 5	Discussion and Analysis
	Chapter 6	Conclusions and Future Work
	Chapter 7	Refection
End matter		References Appendices (Optional) Index (Optional)

Table 3.2: Example of a software engineering-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Requirements specifications
		Analysis
		Design
		Implementations
Chapter 4	Testing and Validation	
Chapter 5	Results and Discussion	
Chapter 6	Conclusions and Future Work	
Chapter 7	Reflection	

3.1.4 Example of a science lab-type main text structure

If you are doing a science lab experiment type of project, you may use the methodology section suggested in Table 3.5. In this kind of project, you may refer to the "Methodology" section as "Materials and Methods."

Table 3.3: Example of an algorithm analysis type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Algorithms descriptions
		Implementations
		Experiments design
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.4: Example of an application type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Problems (tasks) descriptions
		Algorithms/tools/technologies/etc. descriptions
		Implementations
		Experiments design and setup
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.5: Example of a science lab experiment-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Materials and Methods	
		Problems (tasks) description
		Materials
		Procedures
		Implementations
		Experiment set-up
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.2 Example of an Equation in LaTeX

Eq. 3.1 [note that this is an example of an equation's in-text citation] is an example of an equation in LaTeX. In Eq. (3.1), s is the mean of elements $x_i \in \mathbf{x}$:

$$s = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{3.1}$$

Have you noticed that all the variables of the equation are defined using the **in-text** maths command $.$, and Eq. (3.1) is treated as a part of the sentence with proper punctuation? Always treat an equation or expression as a part of the sentence.

3.3 Example of a Figure in LaTeX

Figure 3.1 is an example of a figure in LaTeX. For more details, check the link:
wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.
Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the "Flow algorithm software". Here is the link for that: flowgorithm.org.

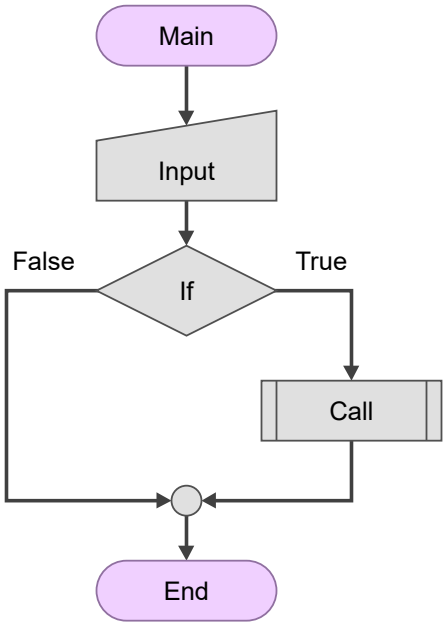

Figure 3.1: Example figure in LaTeX.

3.4 Example of an algorithm in LaTeX

Algorithm 1 is a good example of an algorithm in LaTeX.

Algorithm 1 Example caption: sum of all even numbers

Input: $\mathbf{x} = x_1, x_2, \ldots, x_N$
Output: $EvenSum$ (Sum of even numbers in $\mathbf{x}$)

1: **function** EvenSummation($\mathbf{x}$)
2: $EvenSum \leftarrow 0$
3: $N \leftarrow length(\mathbf{x})$
4: **for** $i \leftarrow 1$ to N **do**
5: **if** $x_i \mod 2 == 0$ **then** ▷ check if a number is even?
6: $EvenSum \leftarrow EvenSum + x_i$
7: **end if**
8: **end for**
9: **return** $EvenSum$
10: **end function**

3.5 Example of code snippet in LaTeX

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 1 is usually preferred over code snippets in a technical/-scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```python
import numpy as np

x  = [0, 1, 2, 3, 4, 5] # assign values to an array
evenSum = evenSummation(x) # call a function

def evenSummation(x):
    evenSum = 0
    n = len(x)
    for i in range(n):
        if np.mod(x[i],2) == 0: # check if a number is even?
            evenSum = evenSum + x[i]
    return evenSum
```

Listing 3.1: Code snippet in LaTeX and this is a Python code example

Here we used the "\clearpage" command and forced-out the second listing example onto the next page.

```java
public class EvenSum{
    public static int evenSummation(int[] x){
        int evenSum = 0;
        int n = x.length;
        for(int i = 0; i < n; i++){
            if(x[i]%2 == 0){ // check if a number is even?
                evenSum = evenSum + x[i];
            }
        }
        return evenSum;
    }
    public static void main(String[] args){
        int[] x  = {0, 1, 2, 3, 4, 5}; // assign values to an array
        int evenSum = evenSummation(x);
        System.out.println(evenSum);
    }
}
```

Listing 3.2: Code snippet in LaTeX and this is a Java code example

```cpp
int evenSummation(int x[]){
    int evenSum = 0;
    int n = sizeof(x);
    for(int i = 0; i < n; i++){
        if(x[i]%2 == 0){ // check if a number is even?
            evenSum = evenSum + x[i];
      }
    }
    return evenSum;
}

int main(){
    int x[]  = {0, 1, 2, 3, 4, 5}; // assign values to an array
    int evenSum = evenSummation(x);
    cout<<evenSum;
    return 0;
}
```

Listing 3.3: Code snippet in LaTeX and this is a C/C++ code example

3.6 Example of in-text citation style

3.6.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more them once.

3.6.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase "Eq" for an equation before using an equation number with (\eqref{.}). Use "Table" to refer to a table, "Figure" to refer to a figure, "Algorithm" to

refer to an algorithm and "Listing" to refer to listings (code snippets). Note that, we do not use the articles "a," "an," and "the" before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence "A report structure is shown in **the** Table 3.1" should be written as "A report structure is shown **in** Table 3.1."

3.7 Summary

Write a summary of this chapter.

Note: In the case of **software engineering** project a Chapter "**Testing and Validation**" should precede the "Results" chapter. See Section 3.1.1 for report organization of such project.

Chapter 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in LaTeX

Table 4.1 is an example of a table created using the package LaTeX"booktabs." do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of liens (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork) goes below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.1.
- The **caption of a Table goes above** the table. See the example in Table 4.1.
- The **caption of an Algorithm goes above** the algorithm. See the example in Algorithm 1.
- The **caption of a Listing goes below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with "Results" Chapter as " Results and Discussion" as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader's understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summaries the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most signification contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Chapter 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in LaTeXand use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planed aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is "**References**," which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between "References" and "Bibliography." The list of "References" strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of "References."

References

Addison Howard, d. (2019), 'Natural language processing with disaster tweets'.
URL: *https://kaggle.com/competitions/nlp-getting-started*

Chanda, A. K. (2021), 'Efficacy of bert embeddings on predicting disaster from twitter data', *arXiv preprint arXiv:2108.10698* .

Deb, S. and Chanda, A. K. (2022), 'Comparative analysis of contextual and context-free embeddings in disaster prediction from twitter data', *Machine Learning with Applications* **7**, 100253.

Fontalis, S., Zamichos, A., Tsourma, M., Drosou, A. and Tzovaras, D. (2023), 'A comparative study of deep learning methods for the detection and classification of natural disasters from social media'.

Iparraguirre-Villanueva, O., Melgarejo-Graciano, M., Castro-Leon, G., Olaya-Cotera, S., John, R.-A., Epifanía-Huerta, A., Cabanillas-Carbonell, M. and Zapata-Paulini, J. (2023), 'Classification of tweets related to natural disasters using machine learning algorithms'.

Phil Culliton, Y. G. (2019), 'Natural language processing with disaster tweets'.
URL: *https://kaggle.com/competitions/nlp-getting-started*

Saddam, M. A., Dewantara, E. K. and Solichin, A. (2023), 'Sentiment analysis of flood disaster management in jakarta on twitter using support vector machines', *Sinkron: jurnal dan penelitian teknik informatika* **8**(1), 470–479.

Appendix A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Appendix B

An Appendix Chapter (Optional)

...