Crime News Generator: A Prototype Based on the NYPD's CompStat DataSet

Abdullah Al-Syed & Jayson Ng
Department of Computer Science
Columbia University
New York, NY, 10027

1 Introduction

In this project, we designed the prototype of a simple computational storytelling system. Computational storytelling is a cutting-edge field that allows algorithms to generate stories based on trusted data sources. Currently, we are in the age of big data, and having such algorithms is extremely useful because they can efficiently process and extract very large datasets at speeds much faster than humans. The applications for such algorithms range from the world of finance to sports. They are also especially powerful in generating stories based on real-time data. With this in mind, we attempted to make an automated crime report generation tool that creates short news snippets based on crime statistics.

2 Overview

Many news stories involve analyzing past trends. For instance, a story on crime in the city would likely focus on how crime rates have plummeted. Rather than require a journalist to spends hours and possibly days getting this data and carving a story out of it, the goal of our system will be to extract this information automatically based on what the journalist is interested in publishing and to generate a story out of it. Our approach was to develop a system that collects and analyzes data provided from the NYPD Crime Statistic data set. We developed several template sentences to have the impact we wanted as well as the accuracy of the information being presented. Afterwards it was a simple concatenation of specific values that determined which template is used in the paragraph. The project's scope is to generate short news stories from data sets in a specific domain. Specifically, we have chosen to use publicly available data sets from NYPD‘s Crime Statistics. In order to better focus our generated reports, we limited our scope to specific crimes such as murders, robberies, felony assaults, rapes, burglaries, general larcenies and automobile grand larcenies, misdemeanor assaults, misdemeanor sex crimes. This way it will allow for a more concise and accurate report. Then we utilize basic statistical measurements to determine positive trends in for each report. After generating the first iteration of the report we will then request a reader to score each sentence for accuracy. The score is then taken into consideration when revising our template sentences. This approach allows us to progressively generate a more accurate report.

3 Related Work

Research in the computational storytelling domain has been done by some of the top schools in the United States, and a recent symposium at the Columbia University Tow Center for Digital Journalism brought together some of the leading researchers and journalists in this field. Companies like Narrative Science are already in this business using big data to generate stories that are relevant and can be understood by ordinary people. There are many uses for computational narrative tools, one example of such a project is the recent story in the LA Times that was generated by a "quake bot" about an earthquake that struck nearby. This automatically generated story incorporated data from several sources such as the US Geological Survey, and although it had to be approved by the journalist before being published, it was ready within 3 minutes after the event. This meant that it allowed the journalist to maximize on the amount of time it took to gather and develop a story for the readers.

4 How Crime News Generator Works

We built an automatic news generation system prototype that produces short snippets of news items relating to specific crime in New York City‘s precincts and boroughs. The dataset we used was the NYPD’s online CompStat statistics that the department uploads weekly [1]. We first used the open-source Tabula platform to generate CSV files that captured the tabular data from the source PDFs as well as additional information such as the precinct and borough information. Our system is written entirely in Java and it thus uses the power of Java‘s regular expressions to extract and store the crime data through the use of object-oriented programming. At a high-level each individual weekly crime report is represented as a Java class, and associated with it are dozens of instance variables that correlate directly to the fields from the source data file, such as the number of burglaries in the past 1 week or the number of burglaries during the same week last year and so on.

In order to generate the news snippet, the system requires the user to provide the names of the source CSV files starting with the precinct file(s) and followed by the borough file (that for the purpose of this demonstration all reside in the same directory as the Java source files) as command-line arguments when running the program. The user is then presented with a menu and a choice as to whether they‘d like to report a news story on falling crime rates or rising crime rates. Naturally, the two are not mutually exclusive, and at any given time, both quantities are constantly in flux; thus, the onus is on the journalist to decide what sort of story they wish to tell about crime. Once the user has selected this option, he or she is further asked to select what sort of crime the story should highlight and make more prominent vis-a-vis other statistics.

The system then proceeds to generate a news story using random number generators to select from predefined template sentences and plugging in statistical data in order to generate a complete news snippet. Naturally, since the text is generated using template sentences and relevant matched data is then substituted in parts of those sentences, we perform post-processing to make the output text look elegant, for instance by auto capitalizing the first words of the generated sentences, removing extra punctuation. Nevertheless, some minor punctuation details were not addresses, such as adding an and after the last comma when a list of crimes is matched, extracted and then substituted in the template sentence. Since the sentences are partially human-generated and partially computer-generated, we constructed a simple evaluation system that lets users rate each individual sentence on a scale to allow future improvements. The next section explains this in more detail.

In sum, we were successfully able to develop a simple computational storytelling system that outputs accurate and concise reports for users. Our system focuses on accuracy and impact. The idea was to have a short but concise and accurate report of crime in specific areas. We believe that in future this tool can greatly benefit civilian awareness and morale for their area. Since it allows users to quickly gather all the relevant information and structure it in a concise and easy to read format. However, we feel that it is possible to add further features and improve upon our current system. Features such as in-depth trend analysis could be developed into our current implementation but our limiting factor for this would be data points.

5 Sample News Snippet

The following is a sample news snippet generated from data in the 40th precinct as well as the overall borough data for Bronx (by providing the filenames cs040pct.sv and cspbbx.csv as command line arguments to the Java program):

Crime statistics are encouraging this week. There was a 66.7 % drop in the number of burglaries in the 40th precinct compared to the same period last year. Felony assaults, housing crimes, misdemeanor assaults also plummeted in the 40th precinct. Looking beyond just the 40th precinct, the Bronx has seen a 6.5 % fall in the total number of burglaries during this week last year.

6 Evaluation System

In terms of interface we created a simple command line rating system where the reader will be shown a series of sentences from the report generated and thus will be tasked with scoring each sentence out of 10 (where 10 being the most accurate and 1 being the least). After which the final score acquired will be considered in the algorithm to create a more accurate report for the next iteration.

In order to better gauge our reports we’ve set the acceptable score for a report to be 7.5. From there we strived to improve the accuracy and impact of each report through the method described before. After each iteration of templates we have a small panel of testers score our automated reports. From the data we've collected from testers, we found that through the use of template sentences we have good control over sentence structure and overall tone. Thus in terms of grammatical accuracy we have a high level of user approval. Where the overall average, for each iteration is 7.85, 7.985 and 8.25 for iterations 1 through 3 respectively. Thus, we concluded that with each iteration the slight modifications to our template sentences have become more accurate and appealing to users.

7 Conclusion and Future Work

Working with regular expressions in Java is quite time-intensive and involved typing a lot of boilerplate code which made the program quite lengthy (around 2000 lines of code). We believe a scripting language such as Perl or Python would be more ideally suited to such a task as opposed to Java for future enhancements. Also, it might be useful to store all this information in a lightweight database if information persistence is required. At the moment, since the datasets were not too large, and this was a prototype, using classes and objects worked out to be fine, but with large datasets spanning years and possibly decades, a database would be essential. The information would then be better extracted using industry-standard SQL queries as opposed to regular expressions. Regular expressions in java, despite their power, still lack many of the strengths in SQL queries such as the ability to select data based on very specific interlinked criteria using a very short and terse amount of code.

References

[1] (2014, Apr. 14). *NYPD CompStat Data* [Online] Available: http://www.nyc.gov/html/nypd/html/crime_prevention/crime_statistics.shtml