Project 1: Approximate String Search for Geolocation of Tweets

Michael James, 390198

August 25, 2014

1 Introduction

Efficient and effective approximate string has had a profoundly positive impact on modern society. Sophisticated approximate string matching techniques have greatly improved the results of the search engines millions of people interact with daily, have revolutionised how various industries do cooperate and even granted the deaf the ability to speak.

The aim of Project 1 is to explore the effectiveness of various string approximate matching techniques in the context of identifying location names present within the body of a Tweet.

2 Input Data Processing

2.1 Data Extraction

A sample of 2,153,222 locations and 3,845,622 tweets from 101,277 users were provided as part of the project. The tweets and locations had to be extracted into separate files, which could be easily worked with, in the context of evaluating the effectiveness and efficiency of the string approximation techniques.

Tweets with the same associated Twitter account user identifier were combined into a single line of the modified Tweets file. A single line of the modified tweets input file consisted of the users Twitter identifier followed by space separated text consisting of all of the users 'Tweets'.

2.2 Simplifications

A requirement of the project specification was to remove all non-alphabetic characters, excluding space, from the body of the tweet text. Therefore, for the locations list to remain consistent with the modified tweet body's all non-alphabetic characters, excluding space, were also removed from the locations list.

Reducing the alphabet size of the locations and tweets resulted in a non-negligible amount of words containing fewer than 3 characters. Words of one to two characters of length were considered to carry little entropy in determining which locations a Twitter user had 'tweeted' and were removed to further reduce processing time.

After thorough inspection of the locations file, significant overlap between the start of many location strings could be seen. To further reduce the sample size locations which had their first two words or more in common were combined into one location containing the common substring.

Processing the locations list resulted in duplicate location entries, the duplicate locations were removed to avoid using time processing said duplicates.

After processing the size of the input tweet file and location file had been significantly reduced (Table 1).

Table 1: Input Data Size Reduction

	Word Count	**Line Count**
Raw Locations Input File	40,342,945	2,153,222
Processed Locations Input File	2,114,363	830,446
Raw Tweets Input File	xxxxx	xxxx
Processed Tweets Input File	38,886,610	101,277

3 String Approximation Algorithms

4 Results

5 Conclusion