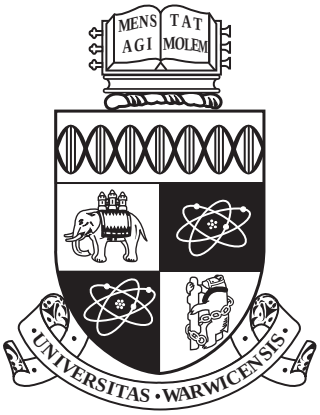


Data mining Twitter for analysis and user comparisons

CS310 - Third Year Project Final Report

Caroline Player
1112108

Department of Computer Science
University of Warwick

2013-2014

Contents

List of Figures

Chapter 1

Introduction

The project documented in this report is about extracting meaning from data made available by the social media Twitter. This report will demonstrate the research which lead to what was ultimately built and the system requirements that are justified by the research, followed by how the project was designed, implemented and managed, an evaluation of the success of the project and finally a conclusion. This chapter outlines the aims and motivations of the project.

1.1 Motivation

A new technology, still in it's infancy, is being developed to gather meaning behind data on the internet. This technology is called the Semantic Web. Currently, one of the biggest projects involved in the Semantic Web is DBPedia, an ontology of the entirity of Wikipedia. By exctracting meaning from such large data sets, a computer can tell you what an article is talking about without it explicitly being told. Another popular research area is sentiment analysis of social media such as Twitter. The aim of which is to understand what users of the social media are talking about at a point in time, or feelings towards a specific topic, and then exploiting that knowledge for individual purposes. These technologies are advancements in artificial intelligence in computing, and they can be further developed to optimise the information we can gain from this data and how it is used.

1.2 Project Aims

The aim of this project is to develop software that can take available data from Twitter and produce a visualisation that displays relevant topics talked about on Twitter and for each user what that topic means to them. The outcome of this project is a unique representation of Twitter users. This sentiment anlysis aims to dynamically understand what is being talked about on Twitter and to who it is meaningful.

Chapter 2

Research

This chapter documents the research that was undertaken for this project. The research shows similar and related work that was used to invistigate the area of unbiased sampling of directed and undirected graphs, sentiment analysis and speech.

2.1 Existing Systems

2.1.1 Alchemy API

2.2 WordNet

A synset is a set of synonyms. WordNet can be interpreted and used as a lexical ontology [Wikipedia]. However such an ontology should be corrected before use since it contains hundreds of basic semantic inconsistencies such as i) the existence of common specializations for exclusive catgories and ii) redundancies in the specialization hierarchy. Furthermore, transformation of WordNet to an ontologyshould involve i) distinguishing the specialization relations subtypeOf and instanceOf relations, an ii) associating intuitive unique identifiers to each category

Main relation among words in WordNet is synonym (as between the words shut and close, or car and automobile). These get grouped into unordered set - synsets. WordNet has 117000 synsets

A hyperonym/hypernym is the super-subordinate relation. It links a word such as furniture, to bed like this {furniture, peice_of_furniture}. This is transitive. If Armchair is a kind of chair, and if a chair is a kind of furniture, then armchair is a kind of furniture. Instances are always leaf nodes in their hierarchies.

Meronymy, the part-whole relation holds between sysets like chair and back, backrest, {seat} and {leg}. Parts are inherited from their superordinates. If a chair has legs, an armchair has legs. They don't go upwards. E.g chair and counter are both furniture, a chair has legs, but furniture doesn't inherit taht upwards because that would mean counter has to inherit it and it doesn't.

Verb synsets are arranged into hierarchies as well. Verbs towards the bottom of the trees express

increasingly specific manners characterizing an event, as in communicate-talk-whisper. The specific manner expressed depends on teh semantic field it's in, in this case, volume. Others are speed: move-jog-run, or intensity of emotion: like-love-idiolize.

Part of speech (POS). Wordnet cosists of four sub-nets, one each for nouns, verbs, adjectives and adverbs, with few cross-POS pointers. Cross-POS relations include the morphosemantic links that hold among semantically similar words sharing a stem with the same maeaning: observe(verb), observant(adj) observation, observatory (nouns).

2.2.1 Classes

Word: getSynset

2.3 Stanford Parser

2.4 Sentiment Analysers

Most sentiment analysers available are aimed at picking at a speicific topic and assessing the feelings towards it.

2.5 Literature Review

2.5.1 Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis

This paper looks at how phrase level sentiment analysis can determine positive and negative expressions [2]. We can pick out keywords in a phrase and look at their **prior polarity**. This is their polarity before we put it in context. This paper discusses new experiments in automatically distinguishing prior and contextual polarity: They use a two step process that uses machine learning and a variety of features.

- First step classifies each phrase containing a clue as neutral or polar.
- Second step takes all phrases marked in step one as polar and disambiguates their contextual polarity.

The system can now automatically identify contextual polarity for a large subset of sentiment expressions.

2.5.2 Combining strengths, emotions and polarities for boosting Twitter sentiment anlaysis

This paper proposes an approach for boosting Twitter sentiment classification using different sentiment dimensions as meta-level features [1]. They combine aspects such as opinion strength, emotion and polar-

ity indicators, generated by existing sentiment analysis methods and resources. The combination provides significant improvement in Twitter sentiment classification tasks such as polarity and subjectivity.

Emoticons can introduce noise [1].

2.6 Tools

2.6.1 Stanford Natural Language Parser

The parser tokenises the text and then tags each token with what it means gramatically i.e a verb, noun, adverb etc. Builds a parse tree for it.

Chapter 3

System Requirements

3.1 Functional Requirements

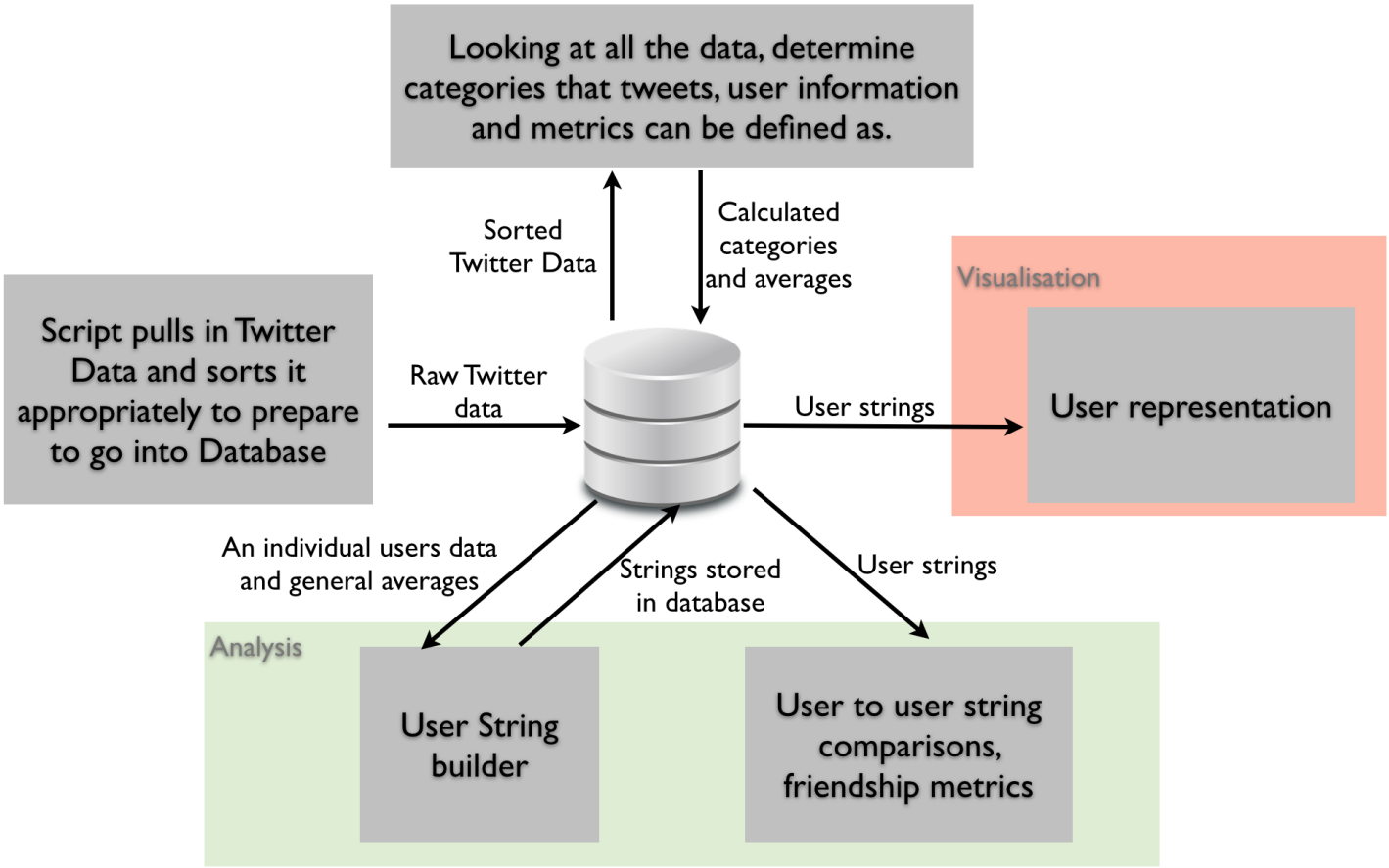
3.2 Non-functional Requirements

Chapter 4

System Design

4.1 System Overview

Here is an image



4.2 Data Collection

The first engine in the system is the data collection. Open source data sets are limited and not necessarily contianing relevant data, as well as possibly being biased towards certain topics, users or geographical locations, the need to produce a system which colleted data more relevant to this project was developed.

4.2.1 Database Design

The idea of the Semantic Web is to give meaning to data, and in the case of Twitter this does not just mean the text in a users tweets. To understand more about a user, we can exploit information on Twitter:

- **Tweets** - useful after having undergone analysis to extract meaning on interests and sentiment towards them.
- **Following** - the types of people a user follows will indicate what they are interested in reading about
- **Followed By** - this is an indication of what they often tweet about. What is important to them that they feell should be expressed
- **Numerical Data** - such as how many friends and followers a user has. If very few they might be a Twitter sheep.
- **Favourites and Retweets** - Giving stronger sentiment to the favourited tweets because they were found particularly important by the user. Or if a user has tweeted a particularly popular tweet then maybe they are more of an authority on this area.
- **Hashtags** - used for emphasis on the object(?) of the conversation. Might give the topic more relavance to the user.

Hundreads of thousands of tweets are stored in the database and so an efficient schema needed to be designed in order to optimise space. This was done by only storing the text of the tweet once with details about it, such as how many times it was retweeted or favourited, by who it was made and it's ID. Then any retweet could simply reference the tweet ID in a separate table. The database needed to be normalised for references. Tweets split into the tweet and it's details. Then a separate table for the users and referencing which tweets they said from the tweet table.

4.2.2 Unbiased Sampling of Twitter

The graph that was being sampled consisted of nodes which were users and edges between them which indicated one relationship of either following or being followed by. When travelling between nodes it therefore makes the more popular users of Twitter much more likely to be visited repeatedly. It is

important to note that if a user is popular and we would like to understand them better to see why they are so popular as this is useful information on what other users are interested in, but not in excess. Visiting many other users that have limited connections is also important. Therefore we can apply what is called the Metropolis-Hastings Random Walk algorithm to try and collect an unbiased sample of Twitter, where we are avoiding the bias of frequently visiting popular Twitter users.

4.3 Analysis

blah

Chapter 5

Implementation

5.1 Data Collection

5.1.1 Tools

The tools used for this were the Twitter API which provides HTTP requests that return JSON objects.

5.1.2 Joshua

When a certain amount of data has been collected from Twitter, a waiting time must be obeyed before you can continue collecting more data. The amount of data that can be collected can be seen in appendix (DO AN APPENDIX ON THE TWITTER API)

5.2 Analysis

5.2.1 Twitter Content

Before users could be given a string which represents whether or not they are interested in certain categories, the categories need to be defined. Determining the categories could not be hardcoded because this would not encapsulate all the topics discussed on Twitter, and lead to a misrepresentation about what is presented. Therefore, what was needed was a system to analyse the text of tweets, and dynamically create the categories, creating data driven results. The following section discusses how the design discussed in Chapter **??**, for categorising the text in tweets was implemented.

Lexical Parsing

The standford parser takes in any amount of text, and tokenises the words. Relationships between words are formed and a tree of the sentence structure can be produced.

Categorising content

WordNet's database of words is limited, and therefore, before a word found in a tweet can be used for categorising, we have to clean it of punctuation, but most importantly, the stem of the word needs to be used. For example, 'Flowers' would raise an exception if searched for in the WordNet dictionary, because it is a plural. So all words being used in analysis are first sent to a method to have the stem of the word found, and the stem is returned and used. This is still subject to errors as often the stemmer provided by the WordNet API returns latin words whose definitions are not stored in it's dictionary. Therefore sanity checks are in place to iterate through all the possible stems of a word and return the first word which applicable that has a definition. WordNet hypernym relation.

5.2.2 User Sentiment Analysis

Chapter 6

Testing

Chapter 7

Project Management

7.1 Data Management

Database snapshots

Chapter 8

Evaluation

Chapter 9

Conclusion

Bibliography

[1] Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '13, pages 2:1–2:9, New York, NY, USA, 2013. ACM.

[2] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.