TÉCNICO LISBOA

HOMOPHILIC SELF ORGANIZING FEATURE MAPS: FINDING TOPICS ON SOCIALY CONNECTED DATA, USING SOCIAL NETWORK RELATIONS

Bernardo Simões

Dissertação para obtenção do Grau de Mestre em

Engenharia de Redes de Comunicações

Júri

Presidente:	Professor Doutor Paulo Jorge Pires Ferreira
Orientador:	Professor Doutor Pável Calado
Vogais:	Doutor whatever full name 2
	Doutor whatever full name 3

Outubro de 2014

Acknowledgments

Remember that your parents have paid for the last 20 something years of your studies and that your advisor had to read this document.

Abstract

Clustering is a widely used technique in data analysis. In this thesis, a generically lgorithm used for clustering is modified in order to enhance the value of socially connected ententies.

To achieve this, we present RubySOM. A framework for easy construction of custom Self-Organizing Maps. With it, it is possible to dinammically change multiple parts of the algorithm, making it extremlly flexible solution to create, train and run custom implementations of the algorithm.

With RubySOM, a relational aware version of the SOM algorithm was created in order to better identify topics on the social network twitter.

Keywords

topic detection, twitter, self-organizing maps, classification, clustering

Resumo

Palavras Chave

detecção de tópicos, twitter, mapas auto organizados, classificação, agrupamento

Contents

List of Figures

List of Tables

List of Acronyms

SOM Self Organizing Maps

NLP Natural Language Processing

U-Matrix Unified Distance Matrix

TF-IDF Term Frequency–Inverse Document Frequency

LDA Latent Dirichlet Allocation

TDT Topic Detection and Tracking

VSM Vector Space Model

IR Information Retrieval

ML Machine Learning

ANN Artificial Neural Network

MDS Multi Dimensional Scalling

PCA Principle Component Analysis

URL Uniform Resource Locator

JSON JavaScript Object Notation

CSV Comma Separated Values

1

Introduction

Contents

With the evolution of social network websites like Facebook and Twitter, the amount of pertinent content about a specif issue is increasing dramatically, which calls for new ways to make sense of and catalog this data. The usage of social networks for branding quality and on-line marketing is specially compelling since 19% of all tweets [12] and 32% of blog posts [24] are about brands or products. Nevertheless, finding topic sensitive information on social networks is extremely complicated due to the fact that documents have very little content, slang vocabulary ,orthographic mistakes and abbreviations. Asur, Sitaram and Huberman [2] successfully predicted box-office revenues by monitoring the rate of creation of new topics based on debuting movies. Their work outperformed some traditional market-based predictors.

Thus academic and enterprise worlds started looking at Machine Learning for new ways to achieve revenue or simply explore and discover patterns in data. As a consequence, the Machine Learning course at Standford is the one with more students enrolling in the year of 2014 [1] with more than 760 students enrolled.

Using unsupervised Machine Learning (ML) Le et al. [19] was able to achieved 81.7 percent accuracy in detecting human faces, 76.7 percent accuracy when identifying human body parts and 74.8 percent accuracy when identifying cats. He used a 9-layered locally connected sparse auto-encoder with pooling and local contrast normalization on a large dataset of images (the model has 1 billion connections, the dataset has 10 million 200x200 pixel images downloaded from the Internet) trained using model parallelism and asynchronous SGD on a cluster with 1,000 machines (16,000 cores) during three days. Even though the amount of computing power used in this project was of several order of magnitude, it is remarkable how an unsupervised algorithm could achieve such results.

Even though a lot of solutions have arisen in order to automate real time searches, topic categorization and many other data intensive tasks, Twitter still uses humans to deliver ads to trending queries, states Edwin Chen's Data Scientist responsible for ads quality at Twitter. On his blog post [2] Edwin Chen describes the process of Twitter to deliver real time adds to trending queries. The main problems that arise in the Twitter platform in order to identify rising topics are:

- The queries people perform have never before been seen, so it is impossible to know beforehand what they mean.
- Since the spikes in search queries are short-lived, there's only a short window of opportunity to learn what they mean.

This means that when an event happens, people immediately come to Twitter in order to know what is happening in a determined place, Twitter solves this issue by monitoring which queries are currently popular in real time, using a Storm topology [3]. After the queries are identified, they

[1] http://www.forbes.com/sites/anthonykosner/2013/12/29/why-is-machine-learning-cs-229-the-most-popular-course-at-stanford/
[2] http://blog.echen.me/2013/01/08/improving-twitter-search-with-real-time-human-computation/
[3] http://storm-project.net/

are sent to a Thrift API [4] that dispatches the query to Amazon's Mechanical Turk service [5] where real people will be asked a variety of questions about the query.

Social Media Analytics is another raising topic that draws from Social Network Analysis [15], ML, Data Mining [30], Information Retrieval (IR) [26], and Natural Language Processing (NLP). As stated by Melville et al. in [24], 32% of the 200 million active bloggers, write about opinions on products and brands, while 71% of 625 million Internet users read blogs and 78% of respondents put their trust in the opinion of other consumers. In comparison, traditional advertising is only trusted by 57% of consumers. This kind of data drives companies to Social Media Analytics as a way to know what people are saying on the web about their companies and products. This new worry has brought to life a lot of new startups like Sumal[6] or ThoughtBuzz[7], but also solutions from the old players like IBM[8] and SAS[9]

Its also important to notice that in the last few years Data Science/Analysis has been a trending topic, mostly due to the fact that big dot-com companies have been having high revenues by exploiting user specific information in order to deliver ads and sell products. Not surprisingly that if you look that in the top 10 ebooks sold by O'Reilly throughout 2013, four are about data science [10].

In this project we will focus on using an unsupervised learning technique based on neural networks named Self-Organizing Maps [17] in order to detect topics in Twitter posts, by using the Social Network users as base neurons for clustering. After the network is trained, it will be possible to categorize tweets in real time. This approach will be better described in subsection **??**.

1.1 Motivation

1.2 Objectives

The main objective of this project is to find topics on tweets by contextualizing the social network involving the person that authored the tweet in the clustering process.

We start by building a dataset, in order to train the Self Organizing Maps (SOM), that will later classify each future tweet that arrives on the network without further delay.

After creating the dataset, we will try to find clusters of topics using the default SOM approach, converting each tweet to Vector Space Model (VSM). After analyzing the results from the default SOM approach, the algorithm will be changed in order to give relevance to the fact that there is a

[4] http://thrift.apache.org/
[5] https://www.mturk.com/mturk/
[6] https://sumall.com/
[7] http://www.thoughtbuzz.net/
[8] http://www-01.ibm.com/software/analytics/solutions/customer-analytics/social-media-analytics/
[9] http://www.sas.com/software/customer-intelligence/social-media-analytics.html
[10] http://shop.oreilly.com/category/deals/bestoforeillydotd.do?code=DEAL&cmp=twnabooksvideosinfo-authornote_best_of_2013

relationship between authors of the tweets.

1.3 Contributions

The main contributions of this work are as follow:

- A method to enhance topic discovery on data with small text corpus and high social significance.
- A framework to easily edit multiple parts of a SOM algorithm, by passing high order functions as configuration.
- To develop highly customizable infinite tweeter crawler that preserves the social network.

1.4 Dissertation outline

Explain how did you organized your thesis.

2

Background

Contents

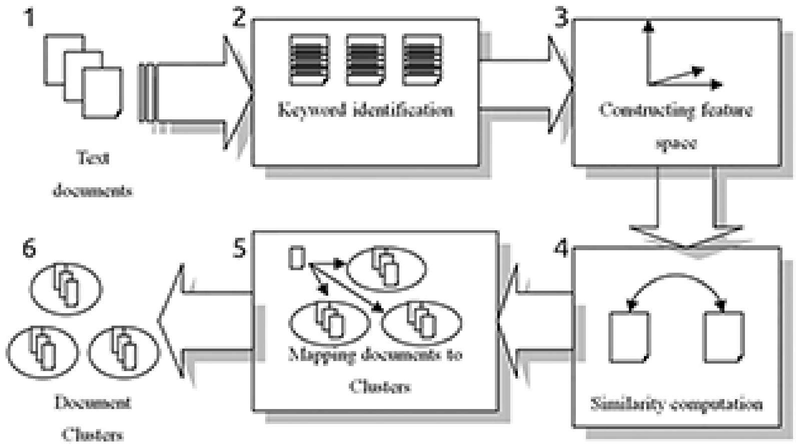

Figure 2.1: Text clustering main framework [9]

In this Section we will start by generally describing what Clustering is and how it works. We will then outline how Self-Organizing maps [17] function, which is the Document Clustering algorithm used on this project.

2.1 Document Clustering

Document clustering is an optimal division of documents into categories without prior knowledge of the data that is being organized, based only on the similarity between them. Due to the fact that no prior knowledge of the data has to be known, Document Clustering is labeled as Unsupervised Machine Learning [10].

Liu et al. [21] asserted that Document Clustering can be used in a variety of Computer Science fields, such as:

- Natural Language Preprocessing.
- Automatic Summarization.
- User preference mining.
- Improving text classification results.

There are two main types of Document Clustering: Hard Clustering and Soft Clustering. In Hard Clustering one document can only belong to one cluster, while in Soft Clustering one document can belong to multiple clusters.

The clustering process usually works as described in Figure 2.1 In the first, step a data set must be provided with the documents, to be clustered. The second step is where non relevant words are removed from the documents, to improve clustering quality [13]. The third step is

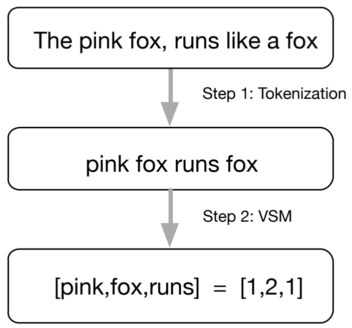

Figure 2.2: Text tokenization and transformation to Vector Space Model.

characterized by converting the keywords of each document into vectors. The most common model used for this task is VSM. In VSM, each vector dimension represents one detected keyword and each document is represented by the vector of keywords in the feature space. This process is ilustrated in Figure 2.2 and works in the following way:

- First step: string tokenization, and token selection. In this case stop words and repeated words will be removed.

- Second step: string to VSM convertion. Each diferent word will correspond to a position in the array, and its value will correspond to the number of ocurrences.

There are many clustering algorithms. In the following section we will describe the particullar case of the SOM, the solution used in our work.

2.2 The Self-Organizing Map

SOM are a two layer, recurrent Artificial Neural Network (ANN) that has the desired property of topology preservation, thus mimiking the way the cortex of highly developed animals brains work. SOM allow cluster visualization of multi-dimensional data, similar to methods such as Multi Dimensional Scalling (MDS) [18] and Principle Component Analysis (PCA) [11] .

As Bação et al. [3] describes, the basic idea behind SOM is to map the data patterns into an n-dimensional grid of neurons, or units. That grid is also know as the output space, as opposed to the initial space, called input space, where the input patterns reside. Both spaces can be seen in Figure 2.4.

SOMs work similarly to the way that is thought that the human brain works. By having a set of neurons that, through learning experience, specialize in the identification of certain types of patterns. These neurons are responsible for categorizing the input patterns for which they are responsible to identify. Nearby neurons will be organized by similarity, which will cause similar patterns to activate similar areas of the SOM. With this topology preserving mapping, the SOM organizes information spatially, where similar concepts are mapped to adjacent areas. The topology is preserved in a sense that, as far as possible, neighborhoods are preserved throughout the mapping process. Output neurons are displayed in an N dimensional grid, generally rectangular, but other structures are possible, such as hexagonal or octagonal. The grid of neurons in the output space, can be divided in neighborhoods, where neurons responsible for the same kind of input reside. In SOM, neurons will have the same amount of coefficients as the input patterns and can be represented as vectors.

Before describing the algorithm it is important to define two key aspects of the SOM, the learning rate and the quantization error. The learning rate is a function that will be decreased to converge to zero. It will be applied to winning neurons and their neighbors in order for them to move toward the corresponding input pattern in progressively smaller steps. Quantization Error is the distance between a given input pattern and the associated winning neuron, it describes how well neurons represent the input pattern. The radius of the neighborhood around the winner neuron is also particularly relevant to the topology of the SOM, deeply affecting the unfolding of the output space as stated by [3].

Algorithm 1: Self-Organizing Map [17]

Data: Input patterns $X = \{\overrightarrow{x_1},\dots,\overrightarrow{x_N}\}$
Result: Trainned map and clustered input patterns
Randomly initialize neurons, $w_i \in \mathbb{R}^D, \forall i$
for $t = 1\ to\ t_{max}$ **do**
  Randomly draw an input pattern, $\overrightarrow{x_d}$
  $p = \arg min_i\{\|\overrightarrow{x_d} - \overrightarrow{w_i}\|\}$
  $\overrightarrow{w_i} = \overrightarrow{w_i} + \epsilon(t) \cdot h_i p(t) \cdot (\overrightarrow{x_d} - \overrightarrow{w_i}), \forall i$
  $\sigma(t) = \sigma_0(\sigma_f/\sigma_0)^{t/t_{max}}$
  $\epsilon(t) = \epsilon_0(\epsilon_f/\epsilon_0)^{t/t_{max}}$

Algorithm 2: U-Matrix

Data: Input patterns $X = \{\overrightarrow{x_1},\dots,\overrightarrow{x_N}\}$, Trainned neurons $W = \{\overrightarrow{w_1},\dots,\overrightarrow{w_N}\}$
Result: U-Matrix
for $t = 1\ to\ N$ **do**
  Randomly draw an input pattern, $\overrightarrow{x_d}$
  $p = \arg min_i\{\|\overrightarrow{x_d} - \overrightarrow{w_i}\|\}$
  $\overrightarrow{w_i} = \overrightarrow{w_i} + \epsilon(t) \cdot h_i p(t) \cdot (\overrightarrow{x_d} - \overrightarrow{w_i}), \forall i$
  $\sigma(t) = \sigma_0(\sigma_f/\sigma_0)^{t/t_{max}}$
  $\epsilon(t) = \epsilon_0(\epsilon_f/\epsilon_0)^{t/t_{max}}$

The learning phase is characterized by the training algorithm 1, which works the following way:

- Neurons can be initialized randomly or it is possible to select a specific initialization.
- Given an input pattern, calculate the distance between the input pattern and every neuron on the network. The euclidian distance 2.1 is generally used.

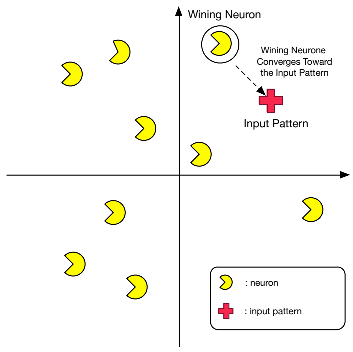

Figure 2.3: Winning neuron converging at learning rate

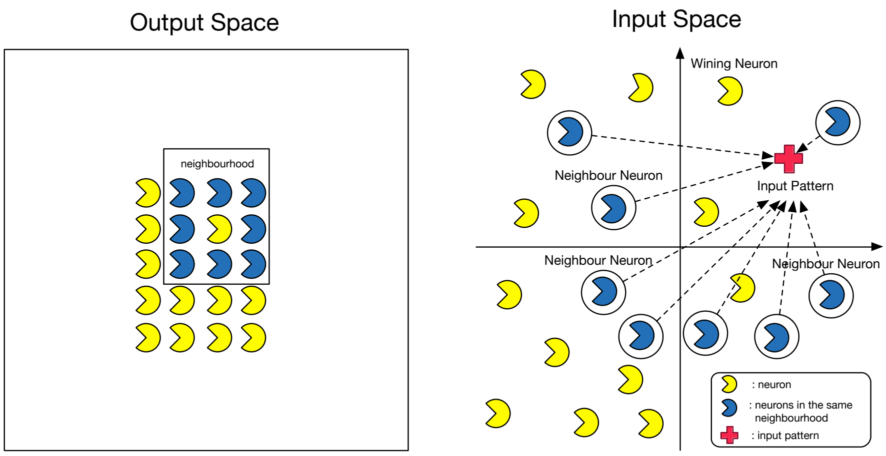

Figure 2.4: On the left the output space neighborhood, on the right the neighbors of the winning neuron converging in the direction of the input pattern

- The winning neuron will be the closest neuron to the input pattern.
- The neuron will move towards the data pattern at a given learning rate, in order to improve his representation as can be seen in Figure 2.3.
- Neighbor neurons will also improve their representation in order to keep the network progressively organized as can be seen in Figure 2.4.

$$Dist = \sqrt{\sum_{i=0}^{i=n} (V_i - W_i)^2} \tag{2.1}$$

In order for the algorithm to converge, the learning rate and the radius of the neighbourhood need to decrease at a given rate. Generally the exponential decay is used. The prediction phase can start after the model is learned. On the prediction phase new input patterns can be quickly assigned to the SOM, without need to apply the learning rate to the winning neuron and his neighbors. Due to the fact that the input pattern will be assigned to the cluster that the nearest neuron is mapping. Thus it very easy and fast to classify new data now.

In order to visually interpret the result of the SOM, Unified Distance Matrix (U-Matrix) method may be used [3]. The U-Matrix is a representation of the SOM in which distances, in the input space between neurons is represented using a gray scale.

The advantages of using SOM is data noise immunity, easy to visualize the data, and parallel processing [21].

3

State of the art

Contents

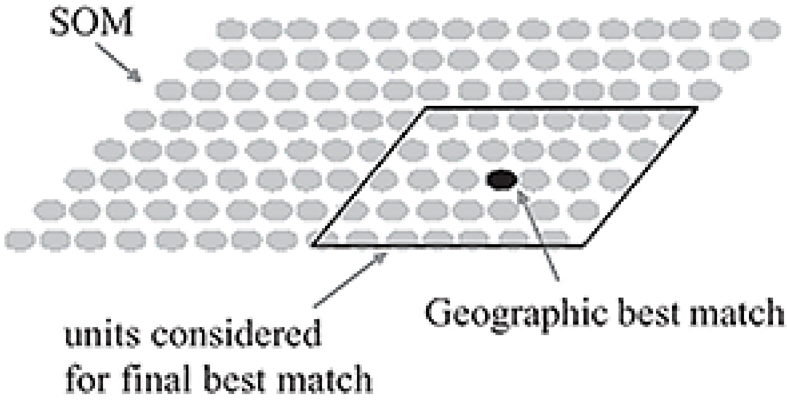

Figure 3.1: Geo-SOM structure, from Bação et al. [3]

This section provides insight of work done in several research areas that are related to the project. In section 3.1 work done using Self-Organizing maps will be described. Section **??** is dedicated to work done on topic detection, in particular on the social network Twitter [1]

3.1 Self-Organizing Maps

Self-Organizing maps are used in a wide are of applications, from authentication systems [9] through network intrusion detection [27] and speech recognition and analysis [16]. We now highlight some of their applications.

3.1.1 The Geo-Som

The Geo-SOM, by Bação et al. [3], applies the first law of geography "Everything is related to everything else, but near things are more related than distant things." to the SOM algorithm. In this case, the winning neuron is chosen in a radius defined by the geographic-coordinates of the data, forcing units that are close geographically to be close in the output space. The representation of the Geo-som can be seen in Figure 3.1.

3.1.2 Detecting Hidden Patterns on Twitter Usage

Cheon and Lee [8] analyzed hidden patterns in the usage of twitter. In this study they started by collecting data from the twitter API of different kinds of topics like "2009 Iran Election" and "iPhone 3.0 OS launch". They made multi level signal extraction not only from information directly present on the tweet, but also by cross referencing with other social websites and with the twitter user profile information. The signals retrieved from the social network can be seen in Table 3.1.

[1] http://www.twitter.com

Table 3.1: Twitter Signals

Twitt Corpus
Tweet Size
Replies
Re-tweets
Hashtags
Presence of URIs and Type of linked content
Type of Device
Tweet Location
Twitter Profile
Account Age
Gender
Country
frequency of posts
Friends to followers ratio
Number of customizations
External Sources
Other Social Network Accounts
Type of website

By applying a SOM, they could find 4 demographical clusters during the Iran 2009 Election. The first cluster was characterized by young web-based Iranians, with twitter accounts not older than 3 months with a high frequency of replies. The second cluster was mainly compound of web users from Iran accounts older that 3 months. The third cluster had Iranian users with mobile clients with large texts clearly trying to raise awareness. The fourth and final cluster represented the users around the world trying to raise awareness about the issue by sharing tweets with URIs. Looking at their analysis about the topic ”2009 Iranian Election” it is clear to see that it was possible to describe the type of users represented in the social network and the way they interact with it.

On the iPhone 3.0 OS launch it was possible to find three main clusters. The first columnster was characterized by male users, accounts older than 90 days, coming from countries where the iPhone is marketed, with high adoption of social media clearly representing the target market of the iPhone or its customers. The second cluster had new accounts with higher rate of followers to followees, high frequency of posts per day, presence of URI linking to technology blogs or websites, no country or gender specified meaning that this cluster was clearly composed by news aggregators and technological news websites. Inside the second cluster there was a sub-cluster of Japanese users which represents the high rate of iPhone adoption in Japan. Finally the third cluster was clearly spammer accounts that where eventually deleted after a couple of months, characterized by popular social connections, posting more than 50 tweets a day with external URIs and the accounts where not older than a day or so.

In conclusion it was possible to detect Twitter usage patterns and specifically detect spammers before they where banned from the social network.

3.2 Topic Detection and Tracking with Clustering

”Topic Detection and Tracking (TDT) research begins with a constantly arriving stream of text from newswire and from automatic speech-to-text systems that are monitoring selected television, radio, Web broadcast news shows. Roughly speaking, the goal of TDT is to break the text down into individual news stories, to monitor the stories for the events that have not been seen before, and to gather stories into groups that each discuss a single news topic” [1]

There have been proposed many TDT techniques. Many of them rely on the Term Frequency–Inverse Document Frequency (TF-IDF) [4] is not particularly adequate for topic detection on Twitter, due to the fact that tweets are very small, often with by typos or slang words and might be written in multiple languages, sometimes on the same tweet. In this subsection we will take a look at multiple methods of topic detection in general and specifically on the Twitter social network.

3.2.0.A Topic and Trending Detection

Cataldi et al. [7] propose a new technique for emerging topic detection that permits real-time retrieval of the most emergent topics expressed by a community on Twitter. Their work applies the PageRank algorithm [25] to the users follower/followee relationship in order to find the most influential users on the network. It then calculates the most trending topics by relating, social influence, word co-occurrence and time frame. In the end, an interface was created where it would be possible to navigate hot topics in a given time frame. Topic labeling was not automatic and was implicit by the time frame of an event.

Weng et al. [29] also used the PageRank algorithm to find the most influential twitter users on a certain topic. However using a different approach, they represent each twitter user as a bag of words comprising of all the tweets that they have posted, and apply Latent Dirichlet Allocation (LDA) [5] in order to find the topics each user is interested in. In the end it was possible to prove that follower/followee relations on twitter are not just casual, but that people actually follow other people to whom they have some resemblance or common interest. This concept is called homophily and will be further explored by this project.

3.3 Twitter Data Mining

In this subsection, we will focus on work done on the Twitter social network in order to leverage insights on how the public data available from the website can be explored.

Table 3.2: Tao et al. [28] tweet characteristics hypotesis versus influence

Hypotheses	Influence of Features
Syntatical	
Tweets that contain hashtags are more likely to be relevant than tweets that don't	Not Important
Tweets that contain an URI are more relevant that tweets that don't	Important
Tweets that are replies to other tweets are less relevant	Important
The longer the tweet is the more relevant it is	Not Important
Semantic	
The more the number of entities the more relevant a tweet is	Important
Different types of entities are of can have different amount of interest to a give topic	Important
The greater the diversity of concepts mentions in a tweet the more likely for it to be relevant	Important
The relevance of a tweet is determined buy its polarity	Important
Contextual	
The lower the temporal distance between a query and the creation of a tweet the more relevant the tweet is	Not Important
The more the number of tweets created by a user the more relevant one of his tweets will be	Not Important

3.3.1 Tweets Implicit Data

Tweet retrieval and analysis is a double edged problem. On one side the tweet is really small, which makes it almost impossible to retrieve any actual sense from it. On the other hand the amount of tweets generated per day is around 140 million[2] wich means that it is very hard to do a deep analyses of the semantics and content of individual tweet, and that only the more appropriate signals should be evaluated. For this reason Tao et al. [28] evaluated how the multiple signals that could be retrieved, directly or indirectly, from the tweet corpus could mean that a tweet is relevant for a determined topic. In his work, Tao presents premises that seem intuitively true and proves they actually are relevant through a comparison of multiple precision and recall values.Their results on feature comparison are summarized in Table 3.2, the first column consists of all the made hypothesis categorized by type, and the second column tells if the data used actually influenced in precision and recall results. Tao also compared result of topic characteristics, concluding that distinction between local and global events as well as temporal persistence proved to not be relevant on relevance prediction.

McCreadie and Macdonald [23] also approached the issue of having very little content on tweets in order to categorize a tweet, and tried to solve it by applying the content of linked URIs into the tweet body in order to improve precision and recall. The best fitting approach was using Field-Based weighting, where for each tweet a new document is created which contains two fields; the terms in the tweet and the terms in the linked document. Afterwards a learning to rank algorithm called PL2F [22] is used against the dataset from Microblog2011 in order to find the

[2]https://blog.twitter.com/2011/numbers

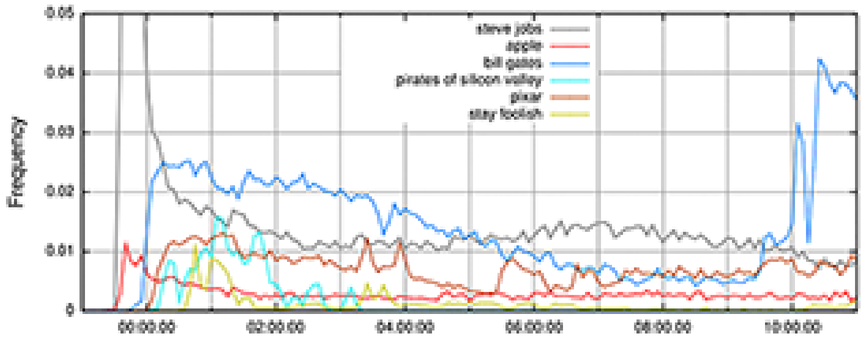

Figure 3.2: The Churn effect: Frequencies of queries related to Steve Jobs death over a 12 hour period in 5-minute intervals, normalized to the total number of queries in the interval. At its peak, the query “steve jobs” reaches 0.15 (15% of the query stream); Graph taken from [20]

best weighting. With this model they where able to improve precision in an order of 0.9, over only analyzing the text contained in the tweets.

3.3.1.A Rapidly Changing Trends

Due to the real time nature of Twitter, using typical retrieval model that relies on term frequency models like Okapi BM25 or language modeling cannot be applied, as stated by Lin and Mishne [20]. The study of topic endurance on the social network proved that topics are presented in bursts of queries and mentions. In addition the typical usage of twitter for search is not the same of Google. When user are searching in twitter they want to find out what is happening right now meaning that classification techniques based on past events cannot respond this kind problem. As stated by Lin and Mishne [20] this problem has not yet been solved at Twitter (or anywhere else at the time of writing this report), and issues a new kind of data analysis approach that was not taken into consideration in the past. This effect of rapidly changing topics and queries based on real time events was named ”Churn”, and can be clearly seen in Figure 3.2.

3.4 Summary

Ending section summarizing the chapter is typically a good idea.

Ensure that the next chapter starts in a odd page

4

Clustering Tweets with Self-Organizing Maps

```
{
  "_id" : { "$oid" : "4fa14bc97e5617025fb14787" },
  "text" : "RT @FastCoDesign: A Paintbrush That Works On The iPad http://t.co/eWjEZAga (@sensubrushman)",
  "id_str" : "197701817864421376",
  "coordinates" : null,
  "in_reply_to_screen_name" : null,
  "in_reply_to_user_id" : null,
  "possibly_sensitive" : false,
  "favorited" : false,
  "in_reply_to_status_id" : null,
  "source" : "<a href=\"http://www.flipboard.com\" rel=\"nofollow\">Flipboard</a>",
  "possibly_sensitive_editable" : true,
  "contributors" : null,
  "retweet_count" : 0,
  "truncated" : false,
  "in_reply_to_status_id_str" : null,
  "geo" : null,
  "in_reply_to_user_id_str" : null,
  "entities" : { Enteties Object },
  "user" : { User object },
  "retweeted" : false,
  "id" : 197701817864421376,
  "place" : null,
  "created_at" : "Wed May 02 14:59:21 +0000 2012" }
```

Figure 4.1: JSON representation of a Tweet.

Tweets cathegorization with **SOMs!** (**SOMs!**) requires a dataset to work on. Due to sistematic API restrictions, which where implemented in order to protect Twitter business model, gathering a dataset that directly maps to the social network is a great chalenge by itself. In order to cathegorize tweets with **SOMs!** first we used a dataset provided by INESC-ID. The dataset had almost 1TB of data in JSON format. Each tweet is comprised of multiple paramters. Figure 4.1 shows how a tweet is represented in JSON format. Inside the tweet there is also information about the user whom created the tweet, shown in Figure 4.3 and enteties shown in Figure 4.2. As can be seen in Figure 4.3 no information about the social relations of the user which emited the tweet are present. Therefor in order to retrieve the social network in which a user is contained, it will be necessary to connect to the Twitter API. Crawling twitter is discussed in further depth in Chapter 5.

In order to better understand the dataset at hand, all the JSON files where converted into CSVin a way to reduce the size of the dataset. While tweets where being converted, Uniform Resource Locator (URL) where removed due to the fact that most of them where minified in order to fit in less that 140 characters. Also, all tweets that where not identified as being in english where also removed. The tweet shown in JSON format in Figure 4.1 is converted to CSV in Figure 4.4.

Identifying tweets that where not in the english language was done through the usage of

```
{ "user_mentions" :
  [{ "indices" : [ 3, 16 ],
    "id_str" : "158865339",
    "name" : "Co.Design",
    "id" : 158865339,
    "screen_name" : "FastCoDesign" },
  { "indices" : [ 76, 90 ],
    "id_str" : "534544812",
    "name" : "Matt",
    "id" : 534544812,
    "screen_name" : "sensubrushman" } ],
  "urls" :
    [ { "url" : "http://t.co/eWjEZAga",
        "indices" : [ 54, 74 ], "display_url" : "bit.ly/IAaKQf",
        "expanded_url" : "http://bit.ly/IAaKQf" } ],
  "hashtags" : [] }
```


Figure 4.2: Enteties mentioned inside the tweet

Ruby library called whatlanguage [1], which tries to identify one language through Bloom Filters. Inside the tweet there is a field which identifies the user language, we found that x is not acurate. Removing tweets that weren't in the english language reduced the amount of different words in x and therefor will reduce the dimensional size of the SOM. The initial dataset features is described table x .

Work done on the INESC twitter dataset with SOMs. SOM implementations used, what where their strong points and weaknesses SVM Dimension reduction and text treatment: compare multiple aproaches to reduce the svm size of tweets without losing relevant information

[1] https://github.com/peterc/whatlanguage

```
{ "default_profile_image" : false,
  "friends_count" : 277,
  "profile_link_color" : "0084B4",
  "followers_count" : 105,
  "url" : null,
  "profile_image_url" : "http://a0.twimg.com/profile_images/690
     647919/19335_266330051104_635886104_3750428_5949465
     _n_normal.jpg",
  "id_str" : "26549290",
  "following" : null,
  "favourites_count" : 7,
  "notifications" : null,
  "profile_background_color" : "C0DEED",
  "statuses_count" : 331,
  "profile_background_tile" : false,
  "profile_background_image_url_https" : "https://si0.twimg.com
     /profile_background_images/79871480/DSC09031_2.jpg",
  "description" : "A Chicago-based designer and educator
     fascinated with the power of design to ignite change \r\n\
     r\n",
  "location" : "Chicago",
  "contributors_enabled" : false,
  "geo_enabled" : false,
  "time_zone" : "Central Time (US & Canada)",
  "profile_sidebar_fill_color" : "cde2e6",
  "listed_count" : 2,
  "profile_sidebar_border_color" : "C0DEED",
  "default_profile" : false,
  "show_all_inline_media" : false,
  "verified" : false,
  "protected" : false,
  "is_translator" : false,
  "profile_use_background_image" : true,
  "profile_image_url_https" : "https://si0.twimg.com/
     profile_images/690647919/19335_266330051104_635886104_3750
     428_5949465_n_normal.jpg",
  "name" : "Karma Dabaghi",
  "follow_request_sent" : null,
  "lang" : "en",
  "profile_text_color" : "333333",
  "id" : 26549290,
  "profile_background_image_url" : "http://a0.twimg.com/
     profile_background_images/79871480/DSC09031_2.jpg",
  "utc_offset" : -21600,
  "created_at" : "Wed Mar 25 17:55:26 +0000 2009",
"screen_name" : "karmadabaghi"  }
```


Figure 4.3: JSON representation of a user, inside a tweet.

```
karmadabaghi,A Paintbrush That Works On The iPad @sensubrushman
```

Figure 4.4: CSV representation of a Tweet. The username is present in the first column and the tweet text on the last.

5

Crawling Twitter for Social Relations

Explain the limitations of the INESC tweet dataset: crawlled by hashtag, social connections can only be obtained through connections to the twitter API, a lot of the tweets had no active users etc.

6

SOM Framework

Explain what got me to create my own ruby library: everybody is making their own SOM algorithms (ex.: websom, hsom etc). Most implementations want to get as close to the metal as possible in order to deliver faster trainings, which makes the lybraries hard to modify. SOM framework is an modular implementation of the SOM algorithm in an higher level programing language which makes it easier to construct and test new SOM algorithms.

7

Homophilic SOM

Describe the alterations made to the default SOM algorithm in order to increase the homophilic (love of the same) relevance while cathegorizing socially connected data.

8

Evaluation Metrics

Contents

- show UMatrixes and multiple steps map trainning of the SOM library trainnig
- show metrics for the crawller, tweets per second, users persecond, size of the dump a long the time.
- compare my som library with other som libraries: training velocity with diferent parameters, map after trainned.
- Compare Homophilic-SOM results with non homophilic: UMatrixes, cluster results, Quantization error, jacknife.

Evaluation of the topic detection on Tweets will be made in two distinct ways. The first way will focus on binary classification using the precision and recall metrics, and will be described in Subsection **??**. The second way will focus on statistically testing the SOM learning process and the computed trained network. This testing process will be described in Subsection **??**.

8.1 Testing for Precision and Recall

Precision and Recall are both ways to measure the rate of right guesses made by the trained SOM network, and are defined in the following way:

- **Precision:** Fraction of retrieved instances that where relevant

$$precision = \frac{|relevant\ documents \cap retrieved\ documents|}{retrieved\ documents} \tag{8.1}$$

- **Recall:** Fraction of relevant instances that where retrieved

$$recall = \frac{|relevant\ documents \cap retrieved\ documents|}{relevant\ documents} \tag{8.2}$$

In order to calculate Precision and Recall we need to have the relevant documents and the retrieved documents. The relevant documents are rather hard to determine because they need to be categorized by humans, which is an expensive task.

8.2 Statistically Testing the SOM

SOM training is always subject to some variability due to multiple causes, like the sensitivity of initial conditions, convergence to local minima and sampling variability, as stated by Bodt et al. [6]. This subsection will present statistical tools to measure the quality of the SOM, by measuring its quantization error and topology preservation.

8.2.1 Quantization Error

The SOM Quantization Error is the mean of all Euclidean distances between the observed data points and their corresponding winning neuron. This value might vary depending on the initialization neurons or the order of the input data fed into the SOM while the training is occurring. When applied to an individual input data, represents how well a neuron is representing input data. Since the SOM Quantization Error represents the mean of all quantization errors from all the input data, generally, the lower the error is the best the SOM was trained.
No general formula exists to minimize quantization error [6] . What is generally done is just to change the number and values of the starting neurons and the order of the input data in order to train multiple SOMs. In the end the SOM with the lowest quantization error is chosen. In this project since multiple approaches to the SOM algorithm and data representation will be tested, as described in Section **??**,and the ones having the lower quantization error will be selected for the prototype.

8.2.2 Topology Preservation

The Self-Organizing Map performs a mapping from the n-dimensional input space into the two dimensional output space and where resides one the most fascinating characteristics, which is that the output map tries to preserve the topology from the input space. This grants the SOM algorithm a way to visualize high-dimensional data that other neural networks or clustering algorithms don't have. Even though this is true, sometimes during training it is not possible to preserve the topology of the network. Thus topology preservation can be measured through the Topographic error Kiviluoto [14] which is the proportion of all data vectors for which first and second BMUs [1] are not adjacent units. In this project the Topographic Error will be calculated for all SOM implementations and VSM usages in order to understand if the representation of the SOM output space is well defined.

8.3 Conclusions

[1] unit that is closest to the winning neuron. BMU Best fitting unit

9
Conclusions and Future Work

Draw your conclusions here and sell your work. Trasmit to the juri how hard it was to develop the presented work.

A future work section is usually here.

Bibliography

[1] Allan, J. (2002). Topic detection and tracking: event-based information organization, volume 12. Springer.

[2] Asur, Sitaram and Huberman, B. A. (2010). Predicting the future with social media. Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 01:492–499.

[3] Bação, F., Lobo, V., and Painho, M. (2005). The self-organizing map, the Geo-SOM, and relevant variants for geosciences. Computers & Geosciences, 31(2):155–163.

[4] Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[5] Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. the Journal of machine Learning research, 3:993–1022.

[6] Bodt, E. D., Cottrell, M., and Verleysen, M. (2005). Statistical tools to assess the reliability of self-organizing maps. Neural networks, 15(8-9):967–978.

[7] Cataldi, M., Di Caro, L., and Schifanella, C. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. In Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10, pages 4:1–4:10, New York, NY, USA. ACM.

[8] Cheong, M. and Lee, V. (2010). A Study on Detecting Patterns in Twitter Intra-topic User and Message Clustering. 2010 20th International Conference on Pattern Recognition, pages 3125–3128.

[9] Dozono, H. (2012). Application of Self Organizing Maps to Multi Modal Adaptive Authentication System Using Behavior Biometrics. Applications of Self-Organizing Maps, pages 120–141.

[10] Hinton, G. E. and Sejnowski, T. J. (1999). Unsupervised learning: foundations of neural computation. MIT press.

[11] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components.

[12] Jansen, B. J., Zhang, M., Sobel, K., and Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. J. Am. Soc. Inf. Sci. Technol., 60(11):2169–2188.

[13] Kang, S.-S. (2003). Keyword-based document clustering. Proceedings of the sixth international workshop on Information retrieval with Asian languages -, 11:132–137.

[14] Kiviluoto, K. (1996). Topology Preservation in Self-Organizing Maps. Neural Networks, 1996., IEEE International ....

[15] Knoke, D. and Yang, S. (2008). Social network analysis, volume 154. Sage.

[16] Kohonen, T. (1988). The 'neural' phonetic typewriter. Computer, 21(3):11–22.

[17] Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE.

[18] Kruskal, J. and Wish, M. (1978). Multidimensional Scaling. Sage Publications.

[19] Le, Q. V., Ranzato, M., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J., and Ng, A. Y. (2012). Building high-level features using large scale unsupervised learning. In ICML. icml.cc / Omnipress.

[20] Lin, J. and Mishne, G. (2012). A Study of" Churn" in Tweets and Real-Time Search Queries (Extended Version). arXiv preprint arXiv:1205.6855.

[21] Liu, Y., Liu, M., and Wang, X. (2012). Application of Self-Organizing Maps in Text Clustering: A Review. Applications of Self-Organizing Maps, pages 205–219.

[22] Macdonald, C., Plachouras, V., He, B., Lioma, C., and Ounis, I. (2006). University of glasgow at webclef 2005: Experiments in per-field normalisation and language specific stemming. In Peters, C., Gey, F., Gonzalo, J., Müller, H., Jones, G., Kluck, M., Magnini, B., and de Rijke, M., editors, Accessing Multilingual Information Repositories, volume 4022 of Lecture Notes in Computer Science, pages 898–907. Springer Berlin Heidelberg.

[23] McCreadie, R. and Macdonald, C. (2013). Relevance in microblogs: Enhancing tweet retrieval using hyperlinked documents. In Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, OAIR '13, pages 189–196, Paris, France, France. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

[24] Melville, P., Sindhwani, V., and Lawrence, R. (2009). Social media analytics: Channeling the power of the blogosphere for marketing insight. Proc. of the WIN, pages 2–6.

[25] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

[26] Salton, G. and McGill, M. J. (1983). Introduction to modern information retrieval.

[27] Samarjeet Borah, A. C. (2013). Intrusion detection system using self organizing map (som): A review. SCIENCE PARK, 1(2).

[28] Tao, K., Abel, F., Hauff, C., and Houben, G.-J. (2012). What makes a tweet relevant for a topic? In Rowe, M., Stankovic, M., and Dadzie, A.-S., editors, #MSM, volume 838 of CEUR Workshop Proceedings, pages 49–56. CEUR-WS.org.

[29] Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). Twitterrank: Finding topic-sensitive influential twitterers. In Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10, pages 261–270, New York, NY, USA. ACM.

[30] Witten, I. H. and Frank, E. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Appendix A

B

Appendix A