Network Analysis on Scraped Fashion-Related Tweets

Lemhour, Oumaima

Award date:
2022

Awarding institution:
Tilburg University

Network Analysis on Scraped Fashion-Related Tweets

JBP000 Bachelor End Project

Oumaima Lemhour

Supervisors:

Joris Mulder
Marlyne Meijerink
Rumana Lakdawala
Giuseppe Arena

Tilburg, July 2022

Abstract

People are increasingly using social media to interact, communicate and share content. The increasing use of social media platforms such as Facebook, YouTube, Instagram and Twitter, generates large amounts of data. Twitter's accessible application programming interface (API) and minimalist concept of microblogging, make the social media platform a good medium for studying online communication behavior. In this project, a network of fashion related tweets is collected for analysis. For the analysis of relations in the network, web scraping techniques are combined with a social network analysis framework. The data is obtained using the web scraping tool Twint, and is analyzed by applying the relational event model (REM) framework. The effects of sets of endogenous variables on the online communication behavior in the retrieved network are analyzed. Resulting models fitted on these variables show good model fit results. In addition, the effect of sentiment is assessed, and found to be significant. Sentiment is analyzed by using the RoBERTa transformers based model. We found that web scraping techniques can be well combined with the relational event model for analyzing such an online communication network. However, dealing with real-world raw data, in combination with a number of limitations, it can be a challenge to retrieve a dataset that represents the network of Twitter users well. The results present how the constructed models fitted this data and how the models can be evaluated.

Contents

Chapter 1

Introduction

The rise of social media has reshaped the way we communicate, to the extent that this is now our preferred form of everyday communication.[17] Users are increasingly using online networks to interact, communicate and share content. People now have multiple accounts on several social media platforms, such as Facebook, YouTube, Instagram, and Twitter. The large amount of users on these platforms generate large amounts of data and digital trails.[14] Data about users and their communication provide unique opportunities to analyze how different kinds of users interact with each other. In comparison to conventional settings, data of digital communication networks are much easier to analyze.[11]

The data as mentioned in the previous section, can be found on Twitter. Twitter is a social media platform where users can follow and engage in conversations about almost everything that is happening in the world. Twitter has a minimalist concept of micro blogging, through which users can communicate with each other using short messages.[9] The platform attracts hundreds of millions of users to personally express themselves, and has a concept of 'following' without reciprocity. There are different ways in which twitter users can communicate with each other, using actions such as: tweets, likes, retweets, threads, tags, and following. Such a big platform with a large amount of different types of users, brings different kinds of conversations, discussions and opinions. There are several methods for sentiment analysis that can bring an interesting view to the analysis of contents of tweets in addition to social network analysis (SNA) methods.[3] Sentiment analysis methods allow for further analysis of the textual contents of tweets and, for example, can tell us whether a specific interaction between users was either positive, negative, or neutral. This behavior is the motivation to take a closer look at the twitter users' characteristic in communication and interactions, based on their behavior on the platform. Twitter has a very open application programming interface (API), which allows programmatic access to tweets for further analysis and research.[20] This combination of Twitter's characteristics and concepts make Twitter a good medium for studying online behavior on the platform.

Web scraping techniques are used to collect data from web pages in an automatic way.[7] Instead of manually extracting information from web pages, web scraping can be used. This is a set of techniques with the goal of looking for certain kinds of information, extracting and aggregating this information into a new web page or database. Scrapers can be used to transform unstructured data into structured databases. Twitter is becoming the the preferred social network for data collection.[10] Data can be scraped from Twitter through the API. To gain access to the Twitter API, users must make a request and deal with things such as rate limits.[13] Rate limits limit the amount of data that can be scraped in a specific time period. There are several packages and libraries designed in different programming languages, like R and Python, to simplify these processes and make interactions with the Twitter API more available for a wider range of users. Methods that are often used to collect data are RTweet, Twint, Twitter API, and Tweepy.[2]

Users on Twitter interact with each other by sending messages, following, retweeting and more. This results in a network consisting of social actors interacting among each other. Social network analysis (SNA) studies the structure behind these type of networks. SNA focuses on both the contents and patterns of relationships in social networks.[22] This method allows the study of social relations and human interactions between individuals, groups, and communities.[21] Understanding the structure of online social networks can provide disciplines such as sociology and marketing with the opportunity to study social networks at a larger scale, relationship patterns and compare behaviors.[15]

We can imagine that in such a big network of online communication, a large amount of topics are discussed, which provides a large amount of data. In this paper, interactions between Twitter users in such a network will be analyzed. A sub network will be filtered out based on keyword selection. Furthermore, the effect of sentiment in online interactions will be analyzed. Analyzing online Twitter interactions in this study attempts to address the following questions: [1] What are variables that influence interactions in a scraped Twitter network?; [2] What is the effect of sentiment in such interactions between Twitter users?; and [3] How can web scraping and SNA techniques be combined to analyze an online communication network?

Chapter 2

Research Methods

2.1 Data

Many social networks contain a lot of content and linkage data, which can be used for research. Content data can contain text, images, video, audio and other types of multimedia data. Linkage data generally represents the graph structure of the network and the connections between actors. A combination of the two can give insights on the dynamics and characteristics in a network.[1] The real-world data that that is used in this project will be collected from Twitter using web scraping techniques.

2.1.1 Data collection - Twint

Tweepy is an example of a web scraping tool that uses the official Twitter API. Despite the public Twitter API, there are some limitations. It requires authentication, by setting up a Twitter developer account and filling out several forms.[12] The official Twitter API also limits the collection of tweets to the last 3200 tweets only. Because of these limitations, Twint is chosen for collecting the data in this project.

Twint, a publicly available Twitter scraping tool written in python, is designed to overcome these limitations. It can fetch almost all tweets, has a fast initial set up and does not require a sign up for a Twitter developer account.

Because of the benefits, data is retrieved using the Twint package. 10,000 tweets are scraped that contained the keyword "fashion". The tweets are filtered by keyword to attempt to retrieve a network with connected actors with the same interest. In this initial data set, there might be more interactions between different actors that are left unobserved, because they do not contain the selected keyword.To solve this problem, after cleaning the data set, tweets for each unique actor in the initial data set are scraped to gather the unobserved interactions. This gives a network of twitter users that have shown interest in the topic "fashion".

The scraped data does not consist of the right structure yet and has to be cleaned and processed to be able to be analyzed using the REM model.

2.1.2 Data cleaning

After scraping the initial set of data, first all tweets are removed that have zero replies and no receiver(s). This measure prevents having single actors in the network that have no connection to another actor. If the data set contains many senders and receivers, but very few events to connect them, finding repetition of events among the same pair of actors over time is unlikely.[18] This repetition of events creates the relational structures that the relational event model can analyze.

Hence, the tweets that are not a reply and have no replies are not of a good use and are discarded.

After removing the "stand alone" tweets, we gather all unique actors in the data set. Again, tweets are scraped, this time for each individual actor separately.

In the Twitter data, it might be the case that an event has multiple receivers. This could happen, for example, when multiple users are tagged or mentioned in a single tweet. To create an event list, a sequence of each interaction between one sender and one receiver, the events with multiple receivers are modified. These events have been fold out into several events with the same sender, but separate outgoing links to the different receivers. In the data set this results in additional rows for one event, and it helps showing the interactions between every pair of actors separately.

What is also important for creating the right input data for the REM model, is that the time of events are well ordered, and do not contain events occurring at the same time. The time is sorted, and when events occur at the same time a very small amount of time, 0.001 seconds, is added to one of the events to make sure they can be distinguished. The time for each event is transformed into the time since the first event in the dataset. Before this transformation the time values were relatively large integers, which causes rounding with scientific notation when the dataset is saved to a csv-file. The rounding causes events to have the same timestamp, which is something that should be prevented, and that is why the time values have been reduced.

From this data set, a list of unique actors that interact with fashion related tweets, can be obtained. For each actor in this list, a maximum of thousand tweets are scraped. Now we collect interactions between actors in the unique list, even when they have not explicitly mentioned the key word "fashion". For this newly scraped data set we go through the same cleaning steps as mentioned before for the initial data set.

Due to memory limitations, the set of actors in the data, needs to be reduced. This is done by looking at the dyads, pairs of two connected actors, that appear more than 22 times. This was found to be the threshold for not going over the maximum limit of number of actors. This threshold, however, only takes the unique pairs of *("sender", "receiver")*. The interactions in which the receiver sends an event to the sender should also be considered to capture a complete conversation between two actors. This does not create a new problem, since it does not increase the number of actors. Looking at the dyads that appear more often also allows to analyze multiple interactions between the same two actors over time. The resulting final dataset consists of 2074 events and 59 unique actors. The structure of the dataset can be found in Figure 1.

2.1.3 Sentiment - RoBERTa

With scraping the data, each full tweet text becomes available. This gives opportunities for analyzing sentiment. How sentiment in previous interactions affects future interactions is one of the interests of this project.

Feelings and opinions are important features for evaluating human actions. Sentiment analysis is a method that focuses on text-based judgements, responses, and emotions, and is often used in fields such as social media analytics to try and understand the viewpoint of a certain audience.[19] In this project, it can help understand how emotions and positivity or negativity in digital communication affect future interactions between the same pair of actors.

Traditional natural language processing (NLP) sentiment classifiers are often based on bag-of-words. Bag-of-words discard the word order and look at the meaning of the individual words.[8] However, sometimes the meaning of an individual word comes from the context and order of words in a particular sentence. RoBERTa is a transformers based model that takes the context around

time	sender	receiver	tweet	sentiment
1202.4580	Heat4Seat	gooddgrrl	@BaadMann1 @gooddgrrl Some women never learn!	0.0
1249.2950	zipcy88	zipcy8888	@hi_haichi @zipcy8888 OMG❤️❤️❤️❤️❤️	2.0
1298.6970	Heat4Seat	gooddgrrl	@gooddgrrl @BaadMann1 @_Sp4nk0_ @FirmSpanker @...	1.0
1298.9934	Heat4Seat	gooddgrrl	@gooddgrrl @BaadMann1 @_Sp4nk0_ @FirmSpanker @...	2.0
1319.2390	sesameellis	annaspargoryan	@annaspargoryan Trying to relax and knowing I ...	0.0
1322.8150	sesameellis	annaspargoryan	@annaspargoryan I've done some half arsed rela...	0.0

Figure 1: An example subset from the eventlist.

each word in the text into account.

The classifier labels the data with '0: Negative', '1: Neutral', or '2: Positive'. It can also give a score for each label, and in a way give a weight for each label for a specific text, such that a ratio of the different labels will be returned. Due to time restrictions and for maintaining simplicity, each tweet in the data set is labeled with either '0', '1', or '2'. In Figure 1 we can see that RoBERTa can also classify emojis as negative, neutral, or positive.

2.2 Relational event model

Relational event models (REM) can analyze any type of continuous-time social interaction data.[16] There are several statistical models available for the modeling of relational dependencies in network data. Three popular statistical models are; exponential random graph model (ERGM), stochastic actor-oriented model (SAOM), and relational event model (REM). The SAOM and ERGM frameworks aggregate relational events into ties, as cross sectional data. The REM allows you to directly model sequences of relational events without having to aggregate them. This does not discard the way in which events are distributed over time and prevents loss of information.[18] A REM is able to include a wide range of cognitive, behavioral, and social processes.[4] REM is different from traditional agent-based modeling frameworks, as this model makes use of event-history analysis to construct models that can be fit directly to data.

The relational event is the key component of this modeling approach. Such an event is defined as a singular event that is generated by a social actor, the sender, and is directed towards one or more receivers. A REM models the probability of a relational event occurring at a certain time or position in the sequence, the event rate. At a certain time t, the event rate determines two things: 1) which social actor will interact next, and 2) when the next interaction will take place.[16]

The data will be analyzed using the R packages remstats and relevent. Remstats computes statistics for fitting the model and relevent fits a relational event model to the event sequence data.

2.3 Model specification

A relational event model can include different kinds of predictor variables.[5] These can be exogenous, such as characteristics of the actors and of the pairs of actors. The predictor variables can

also be endogenous and focus on the way the actors are embedded in the network. In this project a number of variables are selected, in order to see how these effects influence the interactions in the obtained data set. In addition, the variable of sentiment will be added to analyze its effect in the models and whether it improves the predictions.

Baseline. The baseline effects is included to record the event rate for starting a social interaction, and refers to the tendency to interact. It captures the average tendency of Twitter users to start an interaction when all other statistics are equal to zero.

Inertia. Inertia refers to the tendency for pairs of actors to repeatedly interact with each other. It is expected that twitter users that have interacted with each other in the past more often, will be more likely to interact again. We look at the number of previous interactions at time t.

Reciprocity. Reciprocity refers to the tendency for actors in the network to reciprocate past interactions. We look at the number of times that the receiver has reciprocated the interaction before a certain time point t.

TotaldegreeSender. This effect refers to the tendency of actors to send events, if they have sent and received more tweets in the past. We would expect that an actor would be more likely to send events if they have received and sent more events in the past.

TotaldegreeReceiver. This effect refers to the tendency for actors to receive events, if they have sent and received more tweets in the past. We would expect that an actor would be more likely to receive events if they have received or sent more events in the past

Incoming shared partners (isp). The incoming shared partners effect refers to the tendency of dyads to interact, if they have received more tweets in the past that are from the same sender. We would expect that actors that have received more tweets in the past from the same actors, will be more likely to interact with each other.

Outgoing shared partners (osp) The outgoing shared partners effect refers to the tendency of dyads to interact, if they have sent more tweets in the past to the same receivers. We would expect that users that have sent more tweets in the past to the same receivers would be more likely to interact with each other.

OutdegreeSender. This effect refers to the tendency of actors to send events, if they have sent more tweets in the past. We would expect that someone who has tweeted more in the past will be more likely to send a new tweet.

IndegreeReceiever. Refers to the tendency for actors to receive tweets, if they have received more tweets in the past. We would expect that a Twitter user that has received more events in the past, will be more likely to receive more tweets.

Inertia weighted by sentiment. This effect refers to the tendency of users to repeatedly interact with each other, based on the sentiment in the previous interactions. We would expect that users will interact more if they have had more positive past interactions.

All effects are scaled, by standardizing the effect per time point. After scaling the statistics, one unit increase will equal one standard deviation increase. We standardize the statistics, to make the statistics comparable over time and obtain well-behaved model parameters.[16] We look at the intensity of the specific statistics in the past of time point t, as opposed to the raw counts of past events.

As a basis 5 models are set up with every time varying the set of predictor variables, to analyze how these different models fit the data and the different estimates. For each model there will be a second model with in addition the sentiment inertia. It will be interesting to analyze patterns between the four models, but also compare the differences when sentiment is introduced. The structure of the different models can be seen in Table 1. The sentiment inertia is separately added to every model.

	Model0	Model1	Model2	Model3	Model4
Baseline	*	*	*	*	*
Inertia		*	*	*	*
Reciprocity		*	*	*	*
TotaldegreeSender		*			*
TotaldegreeReceiver		*			*
Isp			*		*
Osp			*		*
OutdegreeSender				*	
IndegreeReceiver				*	
Sentiment Inertia	(*)	(*)	(*)	(*)	(*)

Table 1: REM 5 sets of model parameters

All code and necessary data for running the code can be accessed through the following link: `https://drive.google.com/drive/folders/182oBKzPeId9dYVeVy6BavyPZhaRdYiui?usp=sharing`

Chapter 3

Results

3.1 Resulting Data Set

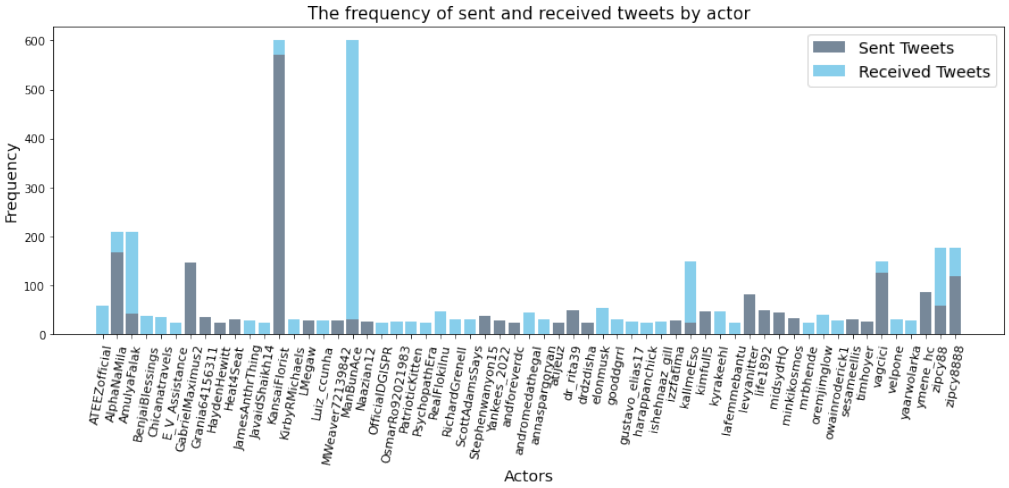

Figure 2: The frequencies of sent and received tweets for each actor.

After retrieving and cleaning the data, we obtain an eventlist as presented in Figure 1 (section 2.1.2). Figure 2 shows per actor the number of tweets they have sent, and the number of tweets they have received. Looking at the total number of interactions for each actor, which can also be explained as the number of times an actor has both sent and received a tweet. This measure has a mean of 70.3, a maximum of 600, and a minimum of 24.

Figure 3 shows the number of times each pair of actors appears in the eventlist, or in other words, the number of times they have had an interaction. We can see a same kind of pattern as in Figure 2, except this time we can see one dyad stands out well above the rest. This frequency count has a mean of 53.2, a maximum of 570, and a minimum of 24. There are also differences in the number of actors each actor interacts with, as presented in Figure 4. We can also see that some Twitter users have interacted with multiple other Twitter actors. However, most of the actors are grouped in pairs.

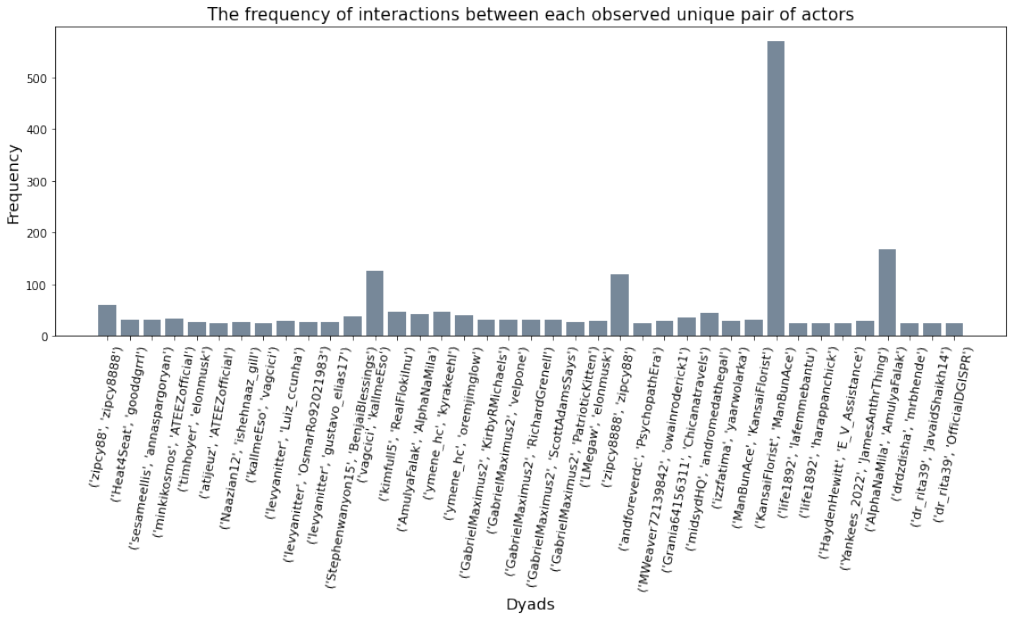

Figure 3: The frequencies of interactions between each observed unique pair of actors.

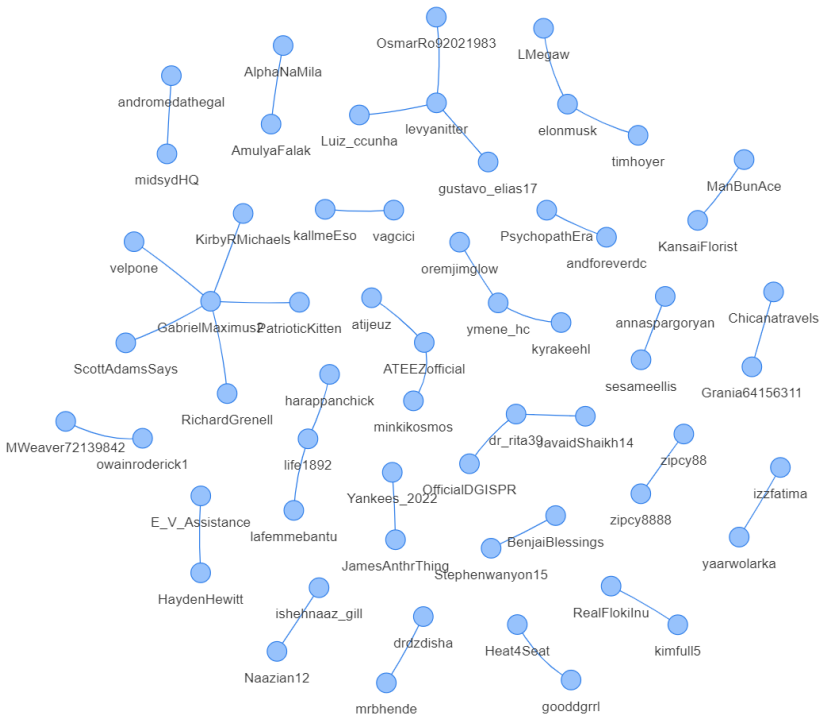

Figure 4: A simple graph visualization of the edgelist.

3.2 Model selection

The five resulting models have different variables and complexities. In order to evaluate the different models, and make comparisons between them, the BIC measure and a goodness-of-fit measure are applied.

The Bayesian Information Criterion (BIC) is useful in selecting a correct model.[6] The measure balances model fit and complexity. Models that are considered better, have a lower BIC value.

Overall, we can see that the BIC decreases as the gof measure increases. We do see that, even though Model 4 without sentiment has a higher gof value, it has a higher BIC value than model 4 with sentiment. If we look at the BIC score, Model 4 with sentiment as weight, is the best model. According to the BIC score, this model has the best balance between model fit and complexity. We can see that when we add the sentiment inertia effect in Model 1 and Model 4, the BIC score improves and decreases.

To evaluate how well the models describe the actual observed event data, the goodness-of-fit measure is used as described in the CONNECT paper.[16] This measure calculates for each event the predicted rates for each dyad. The rate can be compared to the probability of a specific dyad to introduce the next event. The goodness-of-fit (gof) measures the proportion of cases in which the next observed event occurs in the top 1% of dyads with the highest predicted rates.

If we look at the gof measure, we can see that the best two models are Model 2 and Model 4. For each of the models there is no improvement of fit when sentiment is introduced. We can see that there is a big improvement looking at all the models in comparison to the baseline model, which shows that the additional variables do have a significant effect on the output variable.

3.3 Interpretation

3.3.1 Resulting Data Set

We can see that there are some outliers, one actor has sent significantly more tweets than the other actors in the data set, and one actor that has received significantly more tweets. Considering the number of times these two actors appear in Figure 3, it seems like this pair of actors have had many one-sided interactions. In Figure 2, we see that there are a few actors that both send and receive tweets. This could mean that the data set does not contain enough tweets to capture both of these actions for all the users. It could also mean that these Twitter users simply identify more as either a sender or a receiver. For the actors that have only sent tweets in this data set, it is very likely that they have sent unobserved tweets before. In order to be a receiver, a Twitter user must have either posted a tweet, or have been mentioned in a Tweet.

In Figure 4 a graph representation of the data set is presented. The resulting network is not very connected. It seems that Twitter users in this particular fashion-related network are less likely to interact with multiple actors. However, it is questionable whether this is a fair interpretation to make, considering the size of the data set. Had the data set been larger, there might have been a more connected network. The data set is not perfectly balanced. Some actors do not receive events at all and some actors do not send any events. Dealing with real-life raw data, it can be a challenge to retrieve a data set that represents this network of Twitter users well.

3.3.2 Relational Event Model

In Table 2, the results of fitting all the relational event models are presented. We see that all effects are significant in all models, with a p-value below .001. For further interpretation of the network effects, the effects of the model with the best gof measure will be interpreted, that is Model 4.

Since the relational event model is a log-linear model, we can take the log-inverse of the estimated model parameters to obtain a more meaningful metric for interpretation.[16] For the baseline parameter, the log-inverse gives the average number of events per minute for a pair of twitter users,

keeping all the other statistics equal to zero. On average, pairs of twitter users have 6.2846e-7 relational events per minute. Since the events are directed our risk set consists of $59 * 58 = 3422$ potential relational events that can occur among the 59 Twitter users. If we take into account the whole risk set, we can multiply this number by the size of the risk set, and we obtain the average predicted number of relational events per minute, that is 0.0022. By taking the inverse of this, we get that the average expected number of seconds between two relational events is 465 seconds.

The log-inverse of the estimate of the inertia effect gives us that for pairs of Twitter users who interacted with one standard deviation more intensively in the past compared to pairs of users who interacted with average intensity, the baseline rate of starting an interaction is multiplied by 1.42. For these pairs of users the average time between two interactions is 327.66 seconds. So we can see a decrease in waiting time when Twitter users have interacted more intensively with each other in the past.

For pairs of Twitter users that have reciprocated the interactions one standard deviation more intensively than than average intensity, the average expected number of seconds between interactions is again the baseline rate multiplied by the log-inverse of the reciprocity estimate. We obtain that this waiting time is 373.20 seconds. The waiting time decreases as pairs of Twitter users reciprocate more interactions, rather than having a one-sided relation.

Furthermore, for the effects of totaldegreeSender and totaldegreeReceiver, we can see that they have negative estimates. It seems like twitter users that have received and sent more events in the past, keeping all other effects constant, are less likely to send or receive events in the future. This could be dependent on this particular dataset, since it does not capture complete conversations and relationships.

If we generally look at the added weight of sentiment, we can see that it has a positive effect on the event rate. So it seems that the more positive previous interactions between twitter users have been, the more likely they are to start a new relational event.

Effect	Model 0	Model 1	Model 1 Sentiment	Model 2	Model 2 Sentiment	Model 3	Model 3 Sentiment	Model 4	Model 4 Sentiment
Baseline	-8.36 (0.022)*	-10.66 (0.057)*	-10.97 (0.061)*	-13.59 (0.151)*	-12.85 (0.107)*	-9.48 (0.052)*	-9.30 (0.048)*	-14.28 (0.188)*	-13.47 (0.128)*
Inertia		0.69 (0.011)*	0.35 (0.010)*	0.16 (0.001)*	-0.07 (0.009)*	0.54 (0.023)*	0.25 (0.025)*	0.35 (0.009)*	0.08 (0.010)*
Reciprocity		0.57 (0.009)*	0.56 (0.008)*	0.07 (0.003)*	0.06 (0.003)*	0.10 (0.002)*	0.10 (0.002)*	0.22 (0.008)*	0.22 (0.008)*
TotaldegreeSender		-0.78 (0.078)*	-0.28 (0.043)*					-0.54 (0.055)*	-0.26 (0.040)*
TotaldegreeReceiver		-5.14 (0.128)*	-5.92 (0.115)*					-1.49 (0.094)*	-2.00 (0.095)*
Isp				-23.38 (1.408)*	-22.64 (1.362)*			-21.34 (1.465)*	-21.85 (1.332)*
Osp				-150.75 (7.310)*	-111.21 (5.320)*			-161.80 (8.477)*	-109.72 (5.612)*
OutdegreeSender						-0.17 (0.043)*	-0.17 (0.042)*		
IndegreeReceiver						-2.86 (0.166)*	-2.09 (0.146)*		
Sentiment Inertia			0.37 (0.008)*		0.24 (0.009)*		0.19 (0.010)*		0.30 (0.008)*
BIC	38836	25567	25204	22419	22575	28325	28460	21772	21759
gof	18.5%	87.3%	79.6%	87.9%	81.1%	76.8%	71.9%	88.0%	86.9%

Table 2: Relational event model parameter estimates with standard errors, BIC and goodness-of-fit results. (*p < .001)

Chapter 4

Discussion

4.1 Scraped Twitter data

The applied web scraping method, namely Twint, has advantages over other web scrapers. However, regarding documentation, Tweepy's documentation is better when starting off without any scraping experience. It includes tutorials for every step throughout the method. In addition, the data is scraped based on the key word "fashion". Something to take into account, is that this word can have multiple meanings in different contexts. Furthermore, in the data cleaning as a solution for having tweets with multiple receivers, the receivers were fold out into multiple rows. This causes one sender to appear more than once, even when they have sent only one tweet. This changes the representation of the actual data and can influence the statistical inference. Generally, the obtained data set is relatively small. This could cause certain useful patterns being excluded from the analysis. Finally, the code currently in some aspects tailored to this particular data set. Running into errors is manually worked around, so these methods can unfortunately not be applied to any kind of data set.

4.2 Application of Relational Event Model

The obtained models have relatively good model fits. Considering that we have a small data set, the models are kept simple with not too many predictor variables to prevent overfitting. Adding more predictor variables could lead to even better results. The relational event model does not allow for adjusting the risk set to only the observed connected pairs of Twitter users. This causes the size of the intermediate risk set calculations to blow up as the number of actors increases. To solve this issue, the REM model could be adjusted to this particular situation, or the data could be retrieved in such a way that it would not result in a sparse matrix of the risk set. Another solution would be to gain more computational power or memory, such that this task can still be performed with a larger number of actors.

4.3 Strengths and Limitations

Combining web scraping with REM analysis gives the opportunity to analyze different results than would be obtained using experimental data. The use of real-world data can give better insights in how actors behave in a non-experimental setting. However, there are some limitations. The number of actors are limited partly by methods of the REM and by computational power. There are also limitations in the web scraping of Twitter users' characteristics. Twint can provide many characteristics of all tweets, however, with regards to the Twitter users this is very restricted. This also makes less exogenous predictor variables available for this analysis.

4.4 Future Work

For future work, several topics could be taken into account to analyze the differences in networks of Twitter users that are interested in different subjects. In addition, it would be interesting to include more exogenous variables and obtain more characteristics of the Twitter users themselves. However, this might contain sensitive data. Data protection laws and regulations fall outside the scope of this current project, but this is a good thing to consider when processing personal data. Moreover, in this case the selected keyword "fashion" has multiple definitions in different contexts. NLP techniques could be applied to further analyze and filter out the exact tweets that should be targeted for the applicable research.

4.5 Conclusion

This project has shown how web scraping techniques can be combined with network analysis methods in order to gain insights in a fashion-related Twitter social network. As time-stamped relational event data is not always available for social network analysis, in this project it is tested how this type of data can be collected and combined with the REM method, to gain insight into a social network. There are several web scraping techniques with their own benefits. Scraping online relational event data still requires some processing to obtain the right format for analysis with a relational event model. We have found that several variables have significant effect on the event rate in a fashion related twitter network. We also found that sentiment has a significant positive effect on the event rate. We can conclude that Twitter users are more likely to interact with each other when they have had more positive past interactions. Inertia and reciprocity have a positive effect on the event rate. Twitter users are more likely to interact when they have interacted more intensively in the past. Twitter users in the fashion-related network are also more likely to interact, when the receiver has reciprocated more interactions in the past. The remaining endogenous variables; 'TotaldegreeSender', 'TotaldegreeReceiver', 'Isp', 'Osp', 'OutdegreeSender', 'IndegreeReceiver', have a negative effect on the event rate. In this project, it is also shown that collecting, cleaning and getting the data into the right format is a major part of analyzing real-world data.

Bibliography

[1] Charu C Aggarwal. Social Network Data Analytics. *Social Network Data Analytics*, pages 1–15, 2011. 6

[2] Kartikasari Kusuma Agustiningsih, Ema Utami, and Hanif Al Fatta. Sentiment Analysis of COVID-19 Vaccine on Twitter Social Media: Systematic Literature Review. *Proceedings - 2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering: Applying Data Science and Artificial Intelligence Technologies for Global Challenges During Pandemic Era, ICITISEE 2021*, pages 121–126, 2021. 4

[3] Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, 69(1):86–99, 2014. 4

[4] Carter T. Butts. A relational event framework for social action. *Sociological Methodology*, 38(1):155–200, 2008. 8

[5] Carter T. Butts. Revisiting the foundations of network analysis. *Science*, 325(5939):414–416, 2009. 8

[6] Arijit Chakrabarti and Jayanta K. Ghosh. *AIC, BIC and Recent Advances in Model Selection*, volume 7. Elsevier B.V., 2011. 13

[7] R. Suganya Devi, D. Manjula, and R. K. Siddharth. An efficient approach for web indexing of big data through hyperlinks in web crawling. *Scientific World Journal*, 2015, 2015. 4

[8] Piyush Ghasiya and Koji Okamura. Investigating COVID-19 News across Four Nations: A Topic Modeling and Sentiment Analysis Approach. *IEEE Access*, 9:36645–36656, 2021. 7

[9] Martin Grandjean. A social network analysis of Twitter: Mapping the digital humanities community. *Cogent Arts and Humanities*, 3(1), 2016. 4

[10] A Hernandez-Suarez, G Sanchez-Perez, K Toscano-Medina, V Martinez-Hernandez, V Sanchez, and H Perez-Meana. A Web Scraping Methodology for Bypassing Twitter API Restrictions. Technical report. 4

[11] James Howison, Andrea Wiggins, and Kevin Crowston. Validity issues in the use of social network analysis with digital trace data. *Journal of the Association for Information Systems*, 12(12):767–797, 2011. 4

[12] J Roesslein Online] http://tweepy. readthedocs. io/en/v3 and undefined 2009. tweepy Documentation. *Media.Readthedocs.Org*, 2018. 6

[13] Michael Kearney. rtweet: Collecting and analyzing Twitter data. *Journal of Open Source Software*, 4(42):1829, 2019. 4

[14] Anshu Malhotra, Luam Totti, Wagner Meira, Ponnurangam Kumaraguru, and Virgílio Almeida. Studying user footprints in different online social networks. In *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, 2012. 4

[15] Abraham Ronel Martínez Teutle. Twitter: Network properties analysis. *CONIELECOMP 2010 - 20th International Conference on Electronics Communications and Computers*, pages 180–186, 2010. 5

[16] Marlyne Meijerink-Bosman, Mitja Back, Katharina Geukes, Roger Leenders, and Joris Mulder. Discovering trends of social interaction behavior over time: An introduction to relational event modeling: Trends of social interaction. *Behavior Research Methods*, 2022. 8, 9, 13

[17] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J Niels Rosenquist. Understanding the Demographics of Twitter Users. *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 554–557, 2011. 4

[18] Eric Quintane, Guido Conaldi, Marco Tonellato, and Alessandro Lomi. Modeling Relational Events: A Case Study on an Open Source Software Project. *Organizational Research Methods*, 17(1):23–50, 2014. 6, 8

[19] Gagan Reddy Narayanaswamy. Exploiting BERT and RoBERTa to Improve Performance for Aspect Based Sentiment Analysis. 2021. 7

[20] Y. Padma Sai, Deepak Garg, VNR Vignana Jyothi Institute of Engineering and Technology, IEEE Computer Society, and Institute of Electrical and Electronics Engineers. *7th IEEE International Advanced Computing Conference : IACC 2017 : proceedings : 5-7 January 2017, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India*. 4

[21] Karen Stepanyan, Kerstin Borau, and Carsten Ullrich. A social network analysis perspective on student interaction within the twitter microblogging environment. *Proceedings - 10th IEEE International Conference on Advanced Learning Technologies, ICALT 2010*, pages 70–72, 2010. 5

[22] Shazia Tabassum, Fabiola S.F. Pereira, Sofia Fernandes, and João Gama. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):1–21, 2018. 5