The Hitchhiker's Guide to Happiness

An Expedition into a Wholesome Universe

Seam Hasanur
University of Bremen
Bremen, Germany
hseam@uni-bremen.de

Jan Romann
University of Bremen
Bremen, Germany
jan.romann@uni-bremen.de

ABSTRACT

Using the “Happy moments” dataset we trained a topic modelling Latent Dirichlet Allocation (LDA) model using the Gensim library to classify descriptions of happy moments into different categories. Using this classification we built a system that can assign new descriptions of happy moments to one of the identified categories and tries to “recommend” similar happy moments to the people giving input.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; **Information extraction**.

KEYWORDS

LDA, natural language processing, applied machine learning

ACM Reference Format:
Seam Hasanur and Jan Romann. 2020. The Hitchhiker's Guide to Happiness: An Expedition into a Wholesome Universe. In *Datascience 2020: A collection of happy and less happier moments.*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRO

In these days, there are a lot of things that can make you sad. A global pandemic has taken its toll on humanity and the political situation has become very tense in many countries. So, as there are many obvious answers to the question “What makes people *sad* these days?” we wanted to ask a different question: Is it possible to categorize happy moments into topics which can the be utilized in order to predict and recommend equally satisfying experiences?

To answer this question, we used the *HappyDB* dataset [1] which contains over 100.000 descriptions of *happy moments* crowd-sourced by workers on Amazon's Mechanical Turk (MTurk) platform. We chose the HappyDB because we thought it is a very interesting and wholesome dataset which would maybe bring some joy to people by talking about positive experiences. Especially during dire times we wanted to provide a more lighthearted approach to science.

Our model was trained in a way so that it could be able to answer the following research questions:

(1) What is the best way to cluster descriptions of happy moments into fitting topics?
(2) Using our model, how accurately can we predict the topic or category of a happy moment?
(3) Can we use our model as a “recommendation system” for new happy moments to be experienced? If so, how effective can our model be?

In order to answer these questions, our paper is organized as follows: Section 2 outlines the background of our research and the presents the libraries we used for our analysis, before we outline the exact methods and the data used in Section 3. In Section 4 we present our results, and discuss them in Section 5 before highlighting aspects of Fairness, Accountability, Transparency, and Ethics (FATE) in Machine Learning in the context of our project in Section 6. Section 7 concludes.

2 BACKGROUND

HappyDB is a corpus of more than 100.000 happy moments crowd-sourced via Amazon's Mechanical Turk. Understanding the causes of happiness through text-based reflection along with building a recommendation system for new happy moments to experience for people we used popular python libraries and basic NLP techniques.

We have loaded our cleaned_hm.csv and converted the data to pandas DataFrame which offers easier ways to manipulate the datasets. Pandas is a high-level data manipulation tool. It is built on the NumPy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables. In particular, it offers data structures and operations for manipulating numerical tables and time series. In the real world, a Pandas DataFrame will be created by loading the datasets from existing storage, which can be a SQL Database, CSV file or an Excel file.

Numerical Python (NumPy) is one of the core libraries for numeric and scientific computing in the Python Programming Language. This open-source Python library added support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. NumPy is very popular for data analysis and is a part of scientific python.

Processing raw text intelligently is difficult: most words are uncommon, and it's common for words that appear to be totally unique to similar meanings. However similar words in an alternate representation can have completely different translations. In any event, parting text into valuable word-like units can be troublesome

Datascience 2020, April 20–July 27, 2020, University of Bremen, Germany

ACM ISBN 978-1-4503-1234-5/20/07…$9001.00
https://doi.org/10.1145/1122445.1122456

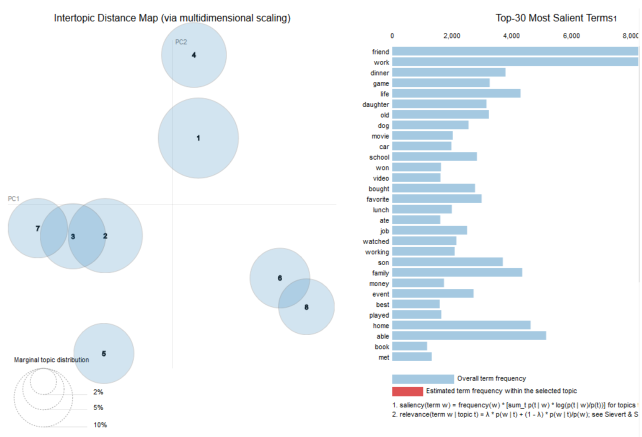
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
8
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms1
0
2,000
4,000
6,000
8,000
friend
work
dinner
game
life
daughter
old
dog
movie
car
school
won
video
bought
favorite
lunch
ate
job
watched
working
son
family
money
event
best
played
home
able
book
met
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & S

Figure 1: Model visualization with pyLDAvis

and difficult in many languages. SpaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. SpaCy helps to know more about the given text corpus and can answer some basic questions, for example: What is the text corpus about? What do the words mean in context? Who is doing what to whom? What companies and products are mentioned? Which texts are similar to each other? And more.

Automatically extracting information about topics from large volumes of texts is one of the primary applications of NLP. We can find the optimal number of topics for a Latent Dirichlet Allocation (LDA) model by creating many LDA models with various numbers of topics. Among those LDAs, we can pick one which has the highest coherence value.

A topic model is a statistical model that is used to separate the documents of a corpus into "topics", i. e. clustering them based on similarity in content. The topic of a document is derived from how frequently words within the corpus occur in a document. One approach for topic modelling is the LDA is one example of such a model. It builds a topic per document model and words per topic model, modeled as so called Dirichlet distributions.

In natural language processing (NLP), the latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics.

PyLDAvis is designed to help users interpret the topics in a topic model, that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization. The visualization is intended to be used within an IPython notebook, but can also be saved to a stand-alone HTML file for easy sharing.

PyLDAvis is an interactive LDA visualization python package. The screen-shoot of pyLDAvis shown in Figure 1 demonstrates how potential results can look like. The area of circles represents the importance of each topic over the entire corpus, the distance between the center of each circle indicate the similarity between topics. For each topic, the histogram on the right side listed the top thirty most relevant terms. LDA helped us extract eight main topics (as can be seen in Figure 1).

3 DATA AND METHODS

The previously mentioned HappyDB dataset can be obtained from kaggle, which is a data science portal offering a wide variety of datasets for users to explore. The HappyDB dataset consists of two CSV files. One of which contains the data we have been working with.

3.1 Data pre-processing

Before feeding the data to our machine learning models, it was important to pre-process the data and and make it more meaningful and informative. For our dataset we have applied a couple of basic pre-processing techniques which are the following:

- Stop Words
- Lemmatization
- N-grams
- Stemming

We first used a function from the Gensim library called `simple_preprocess()` that transformed the string data into a lower case format and removed any punctuation. We then removed stop words, which are strings not containing any relevant information for the classification process, from the dataset. For the removal we imported a list of common stop words for the English language and added a number of domain specific stop words that occurred very often but were not part of the actual description of a happy moment (e. g. “happy”, “happiest”, and “moment”). We then applied Lemmatization which converts words to lemmas, their grammatical base form, whereas Stemming just removes the first or last few characters, often leading to incorrect meanings and spelling errors. This allowed our model to identify similar words even when they are different when it comes to their grammar. We did this by using a `WordNetLemmatizer` from the NLTK library.[1] The pre-processed data could then be transformed into the final input format using the *bag of words* approach, which associates every word in the corpus with a numerical representation. On this basis, each instance of our input data was turned into a dictionary, describing which word IDs were included in the happy moment description. Automatically extracting information about topics from large volume of texts is one of the primary applications of NLP (natural language processing). To process our dataset and develop an automatic happy moment recommendation system, we required an algorithm that can read through these large volumes of text documents and automatically extract the required information and topics discussed in them.

3.2 Role of LDA

LDA's approach to topic modeling is to classify text in a document to a particular topic. Modeled as Dirichlet distributions, LDA builds:

- A topic per document model and
- Words per topic model

[1] In a first approach, we also tried the application of N-Grams and Stemming. However, these techniques produced lower quality results indicated by a lower coherence score of the final models, which is the reason why we only used Lemmatization combined with previously filtering out stop words.

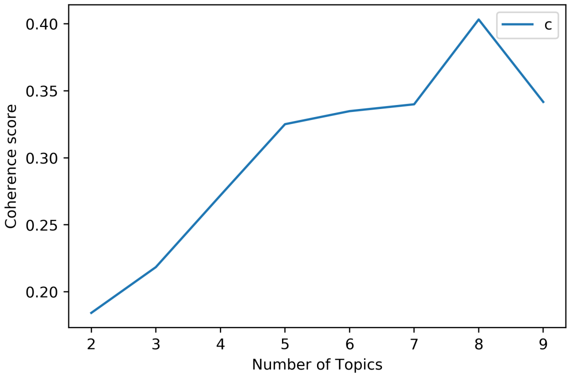

Figure 2: Coherence score by number of topics

For example, According to our trained model, "Today I submitted my final project presentation" the documents topic is "Achievements".

After providing the LDA topic model algorithm, in order to obtain a good composition of topic-keyword distribution, our model rearranged:

- the topic distribution within the document and
- keyword distribution within the topics

While processing, some of the assumptions made by LDA are:

- Every document is modeled as multi-nominal distributions of topics.
- Every topic is modeled as multi-nominal distributions of words.
- We should have to choose the right corpus of data, because LDA assumes that each chunk of text contains the related words.

LDA also assumes that the documents are produced from a mixture of topics.

3.3 Data Collection

As already said previous, the HappyDB [1] is a corpus of more than 100.000 happy moments crowd-sourced via Amazon's Mechanical Turk. Each worker was given the following task:

> What made you happy today? Reflect on the past 24 hours, and recall three actual events that happened to you that made you happy. Write down your happy moment in a complete sentence. (Write three such moments.)

The goal of the corpus is to advance the understanding of the causes of happiness through text-based reflection. The original "kaggle" Dataset is the combination of two different csv files entitled "cleaned_hm.csv" and "demographic.csv". However, for our research work, we have used only cleaned_hm.csv file which has 9 columns.

4 RESULTS

After applying the pre-processing techniques outlined above on the corpus, we identified the appropriate number of topics by calculating an array of LDA models with an increasing number of topics. As can be seen in Figure 2, these calculations revealed that a number of eight topics was the right choice to obtain a maximal coherence score and therefore the best possible separation of happy moments.

We obtained a separation into eight topics that are visualized in Figure 1.[2] As reflected by our relatively low coherence score, there is some overlap between the third and the fourth as well as the second, fifth, and sixth topic.

Based on the words most prevalent in each topic (see Tables 2 to 4 in Appendix A), we assigned the following labels to them[3]:

Topic 1: Achievements (including work)
Topic 2: Joyful moments at home/with family
Topic 3: Friendship/relationships
Topic 4: Unclear/vacation(?)
Topic 5: Eating-related activities
Topic 6: Watching movies and making purchases
Topic 7: Parenting
Topic 8: Playing and watching games (both computer games and sports matches)

As outlined above, we were able to both classify happy moments assign a fitting topic and sort them into one of our labels. For example, the sentence "Today I submitted my final project presentation" was—after applying the same kind of pre-processing to it as we did with our training data—assigned the first topic which deals with achievements and success at work with a probability of about 50%, with the second highest probability of about 26 % being assigned to the eighth cluster, which makes sense as moments where the respective individual or the team they are rooting for are *winning* (and are, thus, achieving their goal) seem to be very prevalent. Also, we can use our model as a kind of recommendation system as we can let it assign a similarity score (ranging from 0 to 1) to the texts in our corpus. With a similarity score of 0.58 each we got the happy moments "I completed a project yesterday and submitted it today" and "Made a decent project presentation" as the two most similar entries from the dataset. However, this already hints at the limitation as a recommendation system as these are *very* similar to the sample moment we used as an input to our model.

Table 1 shows a number of such sample inputs and the assignment made by our model. Comparing these results with the our *interpretation* what each of the topics resembles we can see that our model mostly made the right choice for the topic of the happy moment we described. However, when constructing these sample inputs we noticed that even slight alterations could have a very large impact on the topic selection. For instance, changing the substring "wife" of the penultimate input to "family" resulted in an assignment to the second topic instead of the fourth. This shows that - probably as the pre-processing shortened the already short happy descriptions even more - every single word can have a significant impact on the labelling of a happy moment. It also shows the difficulty of making a clear cut between different kinds of happiness when using an *unsupervised* method as we did.

[2] Using a Principle Component Analysis (PCA) pyLDAvis projects the resulting topics onto a two-dimensional plane which makes it easy to interpret.
[3] As one might notice, these topics actually differ from the ones presented in our final presentation as we managed to improve the pre-processing stage.

Table 1: Assignment of different sample sentences to the determined topics (topic with highest probability was chosen)

Input	Assigned Topic
Today I submitted my final project presentation	1
I've gone on a world trip and traveled to America	2
I had dinner at a fancy restaurant	3
I got promoted and now have a new job	3
I spent some quality time with my girlfriend	3
I went on vacation with my wife	4
I bought a new car	6
I drove my daughter to school	7
I played a video game with a friend	8

Using our model for finding similar moments, however, also worked very well in the cases in Table 1. If this is sufficient for using our model as a recommendation system as we originally intended will be discussed in the next section as well as the question if the LDA has been the right choice for the classification approach we wanted it to use for.

5 DISCUSSION

As we have shown in the last section, we had some success when it comes to training an LDA model for topic modelling of happy moments. In this section, we want to take the discussion we started previously a step further and highlight both the limitations and potentials of our approach.

Using the LDA tools from the Gensim library, we we indeed able to train a model for determining different kinds of happiness within the HappyDB dataset. As intended, it is also possible to use our model for predicting and "recommending" topics based on a given input. However, the still low coherence score of 0.4 (on a scale from 0 to 1) is an indicator of at most mediocre quality, which is in part supported by the tests we performed, using sample input strings. As Tables 2 to 4 show there seems to actually be some issues with the process of Lemmatization ignoring verbs which should be fixed in further research. However, as our experiments with Lemmatization, *n*-grams and Stemming during the pre-process stage have shown, it seems to be a fact, that many descriptions of happy moments are actually very short and loose a lot of information when pre-processed top much. This might be a valid reason to actually consider using different kinds of machine learning approaches when it comes to the classification of happy moments.

We also came to the conclusion, that trying to recommend happy moments based on similarity might not be the best approach to do so, as the happy moments that were retrieved from the corpus were almost always extremely similar to the input string. Furthermore, happiness is a very subjective emotion that can vary based on the person, their current mood or what happened to them recently. For example, recommending to go outside for lunch with your colleagues might not be the best idea, even though they had a very pleasant time when they did last week. Maybe the person is not feeling well today, or they had an argument with one of their colleagues. So we have to take more factors into consideration other than similarity, as even the exact same moment can not always reproduce the exact same feelings. When trying to build an actual recommendation system, one should therefore either focus more on the identified categories and draw a similar, but still different happy moment from the dataset, or use other techniques. The fact, that the dataset contains multiple happy moments from the same person (as each participant was asked to name three moments they experienced), could be used to build a recommendation system that is more similar to the ones that can be encountered in online shops.[4] These thoughts should be kept in mind when developing further research on the topic of happiness and when building applications with the purpose of spreading happiness.

Before coming to the concluding remarks of our project, we want to further reflect upon our work in the aspects of fairness, accountability, transparency and ethics (FATE) of machine learning in the following section.

6 FAIRNESS, ACCOUNTABILITY, TRANSPARENCY, & ETHICS

Separating happy moments into topics might seem like an endeavour that does not bear much potential to cause harm, or is problematic in any way. But at second glance it becomes clear, that techniques, like the ones we used, can be applied in contexts where they could serve as a tool to, for instance, automatically identify what people are talking about on a social media platform. If you use a model that was trained beforehand, you could even perform this analysis almost in real-time.

As with many technologies and scientific results, a so called "dual-use" problem unfolds: An LDA model like the one we trained, could be used for an entirely civilian purpose, like pre-sorting incoming tweets or messages of the Twitter account of a company. However, it can also be used for censorship and surveillance as, to stick with the Twitter example, tweets that are critical of the government could be blocked automatically and those who wrote the tweets could be added to a database of "subversive elements". Especially in China, censorship on social media becomes increasingly automated and is an extreme example of the capabilities for wielding machine learning approaches as a weapon to exercise control.[5] Therefore, as innocent as our happy moments project might seem, the technologies we used have the potential to inflict harm if used unethically in a different context.

Apart from the potential abuse as described above, it is also important to keep in mind that, when building a recommendation system based on our model, the crowd-sourced happy moments are heavily biased towards an American understanding of happiness, as the majority of asked individuals came from the United States while a minority of about 30 percent came from India. This means that our model can not serve as a universal recommendation system for the whole world. While there probably is a certain universal perception of happiness when it comes to relationships with family

[4] "Persons who enjoyed this happy moment also enjoyed 'Eating ice cream with my best friend'." could be a recommendation made by such a system.

[5] As an example to get an overview of the situation in China see the annual "Freedom of the Net" report published by the NGO Freedom House [2].

and friends, the values and expectations within a society most certainly influence the concepts of happiness people strive to fulfill. Therefore, there probably are types of happiness in the world our model will not be able to identify. Also, the recommendations given by our model as a response to an input, will not necessarily be appropriate for a person with a non-American or non-western background. If we want to create a system that can bring happiness to as many people as possible, we need to acquire more data from more diverse backgrounds.[6]

Especially if someone should pursue our original goal of building a web application, it is also very important to consider issues of privacy and data protection, since happy moments are be very personal and sensitive information, that should not be shared in a non-anonymous way with the public. Also, it should be outlined in detail, how the data is processed to ensure transparency.

7 CONCLUSION

In the project presented in this paper, we trained an LDA model for topic modelling of happy moment descriptions derived from the HappyDB dataset [1]. We were able to achieve a reasonable separation into eight topics, which turned out to be the best number of topics, applying the elbow criterion on the coherence score as a metric for determining the best number of topics. But when we reviewed the topics created, we noticed that there is still room for improvement, as at least one of the topics included only very generic terms, with little actual information about the happy moments the respective person has experienced.

The generated model can be used as intended to predict the topic a given happy moment is most likely to belong to. At the same time it lists the moments in the corpus that it perceives as being most similar. However, this prediction is not very accurate as a test with a number of sample inputs showed. Further research should address this issue and should evaluate not only the parameters for the prediction itself, but also the techniques applied during the pre-processing state. Also different approaches besides LDA should be tried out, in order to determine and choose the best approach after comparing the results.

Besides academic research, our approach can be used in similar scenarios with the possibility of embedding them in actual applications. The techniques applied and discussed above can be considered for a number of use cases, like recommending books or products based on actual or descriptive text. However, as we discussed in the section on the FATE of machine learning, any form of application needs to be carried out in a responsible manner as even a project as harmless as ours could be turned into a tool for surveillance or censorship. Further research should also address the likely biases that arise from sampling mostly US-based workers through Amazon MTurk.

So there is still a lot of work to do and a lot to keep in mind, in order to actually reach our goal of bringing more happiness from the world of machine learning to the real one. But after working on this project, we are confident that the expedition into a more wholesome universe will turn out successful in the end.

[6] Note that there is also a selection bias from using data acquired from MTurk as people that have a higher affinity for technology are probably more likely to register there than people who are not. This probably correlates with an age bias, as elderly people are less likely to be experienced with using digital media.

REFERENCES

[1] Akari Asai, Sara Evensen, Behzad Golshan, Alon Halevy, Vivian Li, Andrei Lopatenko, Daniela Stepanov, Yoshihiko Suhara, Wang-Chiew Tan, and Yinzhan Xu. 2018. HappyDB: A Corpus of 100,000 Crowdsourced Happy Moments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan. https://www.aclweb.org/anthology/L18-1103

[2] Freedom House. 2019. *Freedom on the Net 2019. The Crisis of Social Media.*

A MOST RELEVANT WORDS FOR EACH TOPIC

Topic	Term	Term Frequency
1	work	0.070
	able	0.022
	working	0.016
	morning	0.014
	hour	0.014
	finally	0.013
	night	0.013
	finished	0.012
	ve	0.011
	sleep	0.010
2	family	0.026
	home	0.025
	dog	0.023
	came	0.017
	event	0.015
	enjoyed	0.012
	trip	0.012
	nice	0.012
	weekend	0.012
	party	0.011
3	friend	0.123
	old	0.032
	best	0.018
	year	0.016
	met	0.015
	phone	0.015
	brother	0.014
	birthday	0.014
	talked	0.012
	seen	0.012

Table 2: Term distribution within Topics 1–3

Topic	Term	Term Frequency
4	life	0.047
	like	0.016
	people	0.013
	thing	0.013
	know	0.012
	feel	0.011
	dad	0.011
	summer	0.010
	vacation	0.010
	make	0.009
5	dinner	0.052
	lunch	0.025
	ate	0.022
	favorite	0.016
	delicious	0.016
	night	0.014
	movement	0.014
	restaurant	0.013
	coffee	0.013
	nice	0.013
6	movie	0.028
	car	0.028
	bought	0.027
	money	0.022
	watched	0.021
	job	0.020
	able	0.015
	received	0.014
	store	0.014
	food	0.013

Table 3: Term distribution within Topics 4–6

Topic	Term	Term Frequency
7	daughter	0.045
	son	0.027
	school	0.026
	mother	0.016
	mom	0.015
	shopping	0.014
	said	0.012
	birthday	0.011
	child	0.011
	love	0.011
8	game	0.053
	won	0.026
	video	0.026
	favorite	0.021
	played	0.019
	book	0.018
	watch	0.017
	playing	0.016
	team	0.014
	afternoon	0.013

Table 4: Term distribution within Topics 7–8