



TOPIC MODELING REDDIT POSTS

The background of the slide is a collage of Leonardo da Vinci's sketches. It includes architectural drawings of domed buildings, mechanical diagrams of gears and a spiral, anatomical sketches of a horse's head and a human figure, and various floral and plant drawings. The sketches are rendered in a light brown, sepia tone, typical of his original drawings.

REDDIT

- ❖ ** American social news aggregation, web content rating, and discussion website*
- ❖ *Founded June 23, 2005*

REDDIT



Home to microcultures:

- | | |
|-----------------------|--------------------|
| ❖ Academics | ❖ Gamers |
| ❖ Movie Geeks | ❖ Foodies |
| ❖ Political Activists | ❖ Gym Rats |
| ❖ Music Lovers | ❖ Meme Aficionados |
| ❖ Deviants | ❖ & More |

- * As of February 2018, Reddit boasted:
 - ❖ 6th most visited website in the world
 - ❖ 4th most visited in the US
 - ❖ 234 Million unique users
 - ❖ 542 Million monthly visitors
 - ❖ 82+ Billion pageviews
 - ❖ 73+ Million submissions/ posts
 - ❖ 725+ Million comments
 - ❖ 16 Minutes on Site Daily per user
 - ❖ Marketers seek your attention

GOOGLE CLOUD PLATFORM & DATA



Google BigQuery



**Google
Compute
Engine**



- ❖ 321 GB of Reddit posts available on BigQuery
- ❖ Extracted a subset of self posts from June of 2016 with filters for good data
- ❖ Nearly 150k posts to explore in a cloutop powered by 14 vCPU's and 128 GBs RAM

PROCESS

❖ Setup

- ※ Google Cloud - CL configurations (port, account, access, etc.)
- ※ BigQuery - API

❖ Exploratory Data Analysis

- ※ Build intuition about text sample (subreddits, lengths, etc.)
- ※ Looked for unusual spikes to rid sample of bot posts
 - NULL, de (Denmark), test, tr (Turkey), A6XHE (?), es (Spain), removalbot, tldr (Too Long; Didnt Read)
- ※ Could spend many more hours for refined results

PROCESS

- ❖ Preprocessing presented room for creativity from query to data cleaning
 - ✖ Scrub data of non-alphanumeric + links
 - ✖ Remove stopwords
 - ✖ Tokenize with NLTK
 - Count + Term Frequency-Inverse Document Frequency Vectorizers
 - ✖ Bot activity and other noise required a couple passes of data collection and preprocessing

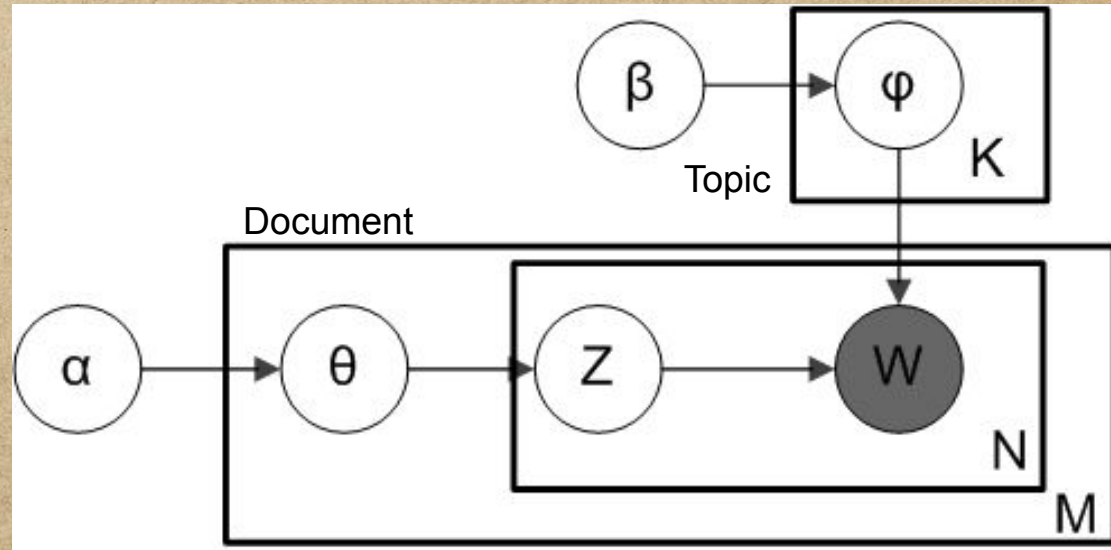
PROCESS

❖ Singular Value Decomposition / Latent Semantic Analysis (SVD / LSA)

- ❖ Inexplicable formula for topics (+ / -)
- ❖ Poor results with TF-IDF, not enough info after SVD
- ❖ K-Means Clustering
 - K-Means could not find meaningful clusters
 - Silhouette scores suggested more and more clusters would result in higher average score
- ❖ Latent Dirichlet Allocation (LSA with sparse non-uniform Dirichlet dist.) for soft clustering and spatial reduction

LATENT DIRICHLET ALLOCATION (LDA)

- ❖ Assumes document generated in a particular fashion
- ❖ α plate: prior for topic distribution in a document
- ❖ β plate: prior for word distribution in a topic



LATENT DIRICHLET ALLOCATION (LDA)

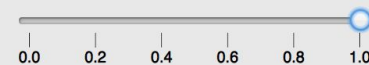
- ❖ LDA gave distributions on 15 topics for each document
- ❖ Final model trained in just over 5 hours with 100 epochs
- ❖ Map of documents -> topics
- ❖ Collection of topics per user

PY LDA VIS

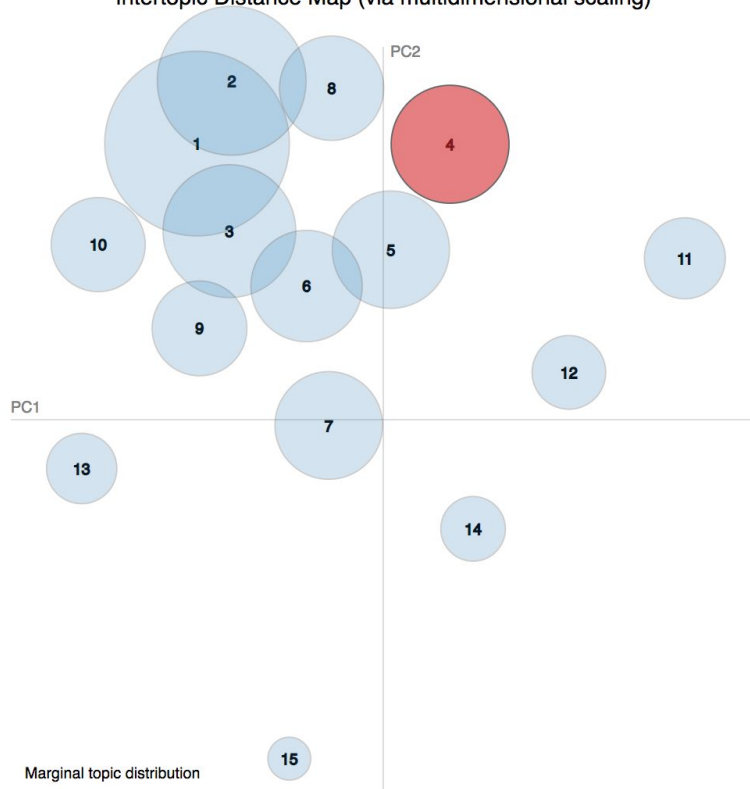
Selected Topic: 4

Slide to adjust relevance metric:⁽²⁾

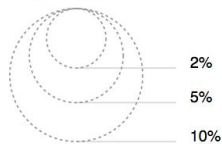
$\lambda = 1$



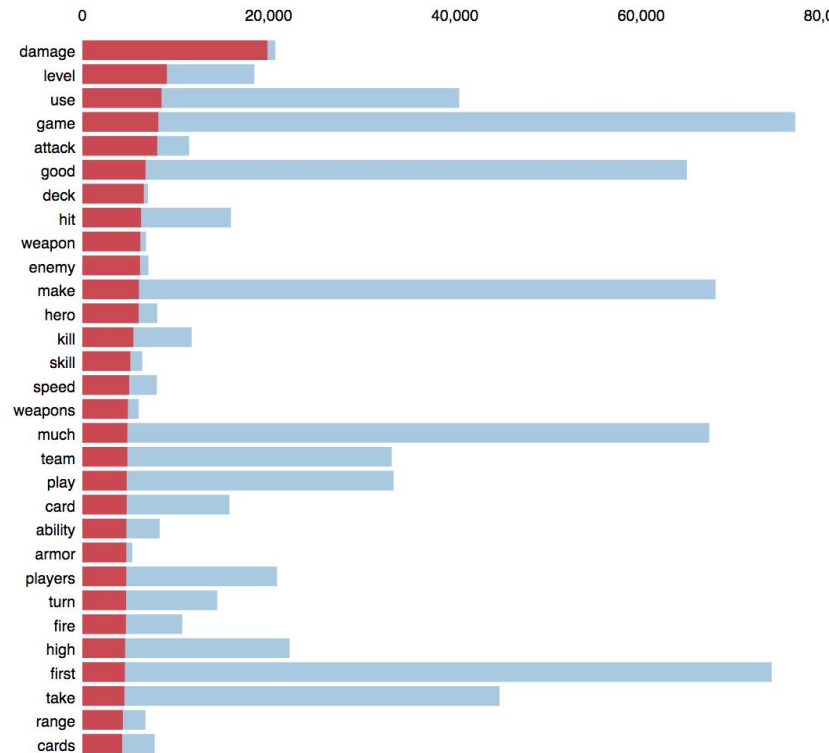
Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution



Top-30 Most Relevant Terms for Topic 4 (7.9% of tokens)



Overall term frequency

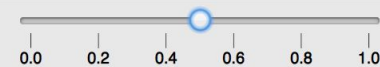
Estimated term frequency within the selected topic

1. saliency($\text{term } w$) = frequency(w) * $\left[\sum_t p(t | w) * \log(p(t | w) / p(t)) \right]$ for topics t ; see Chuang et. al (2012)
2. relevance($\text{term } w$ | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t) / p(w)$; see Sievert & Shirley (2014)

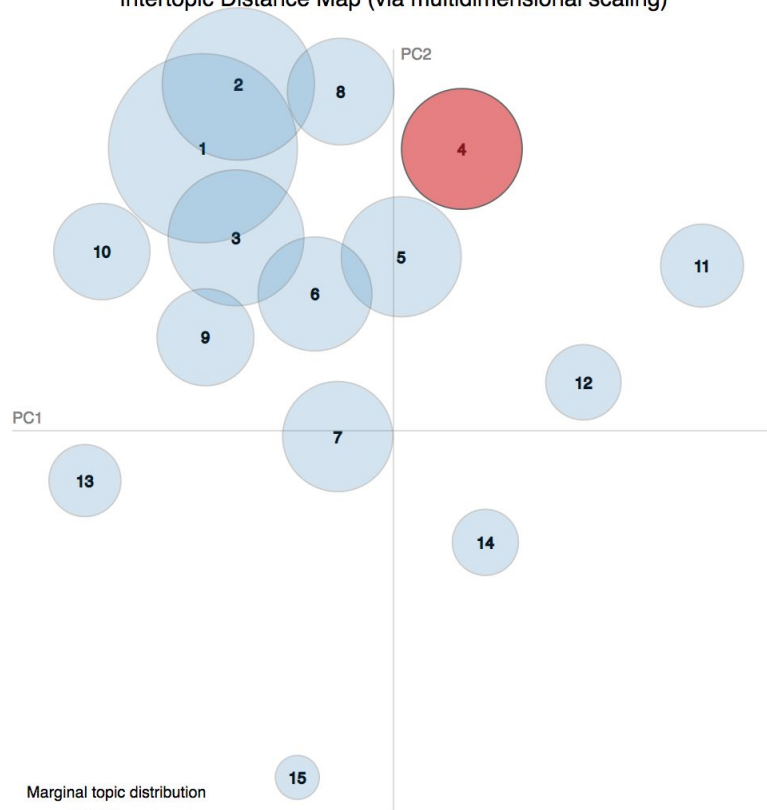
Selected Topic: Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:⁽²⁾

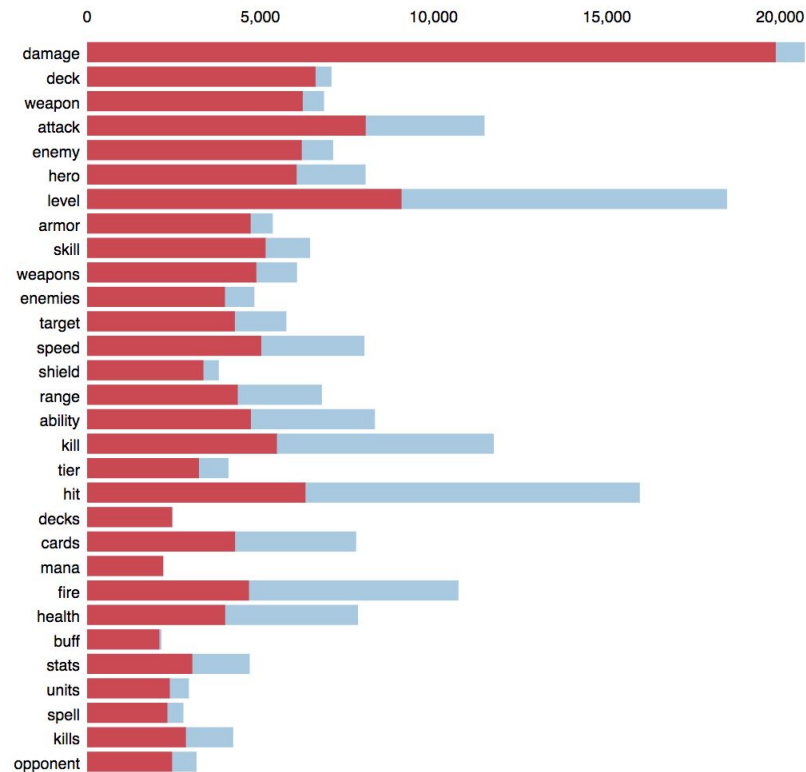
$\lambda = 0.5$



Intertopic Distance Map (via multidimensional scaling)



Top-30 Most Relevant Terms for Topic 4 (7.9% of tokens)



Overall term frequency

Estimated term frequency within the selected topic

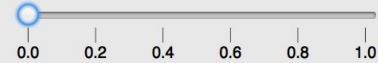
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))]] for topics t; see Chuang et. al (2012)

2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

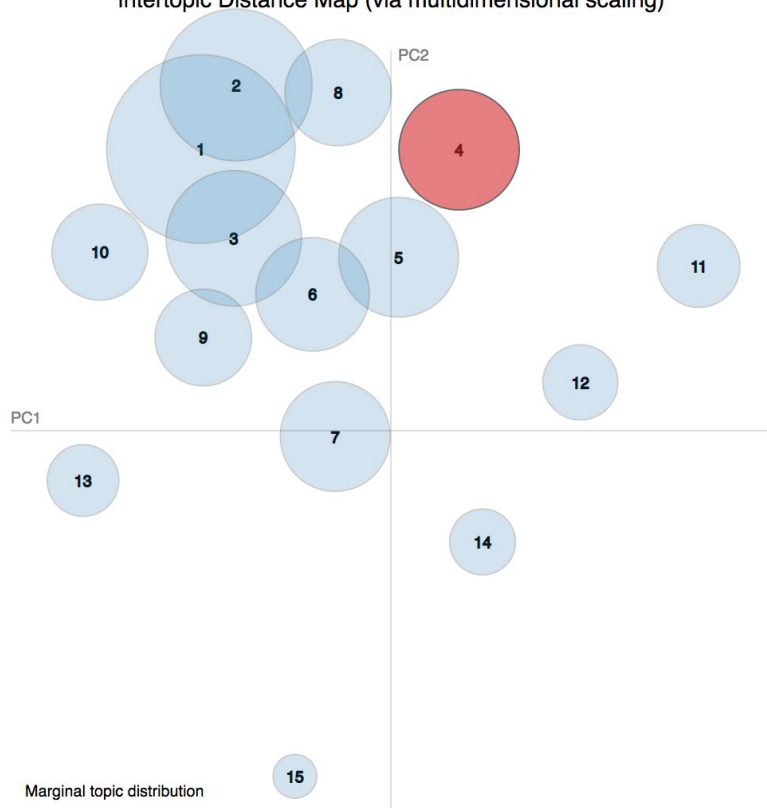
Selected Topic: 4

Slide to adjust relevance metric:⁽²⁾

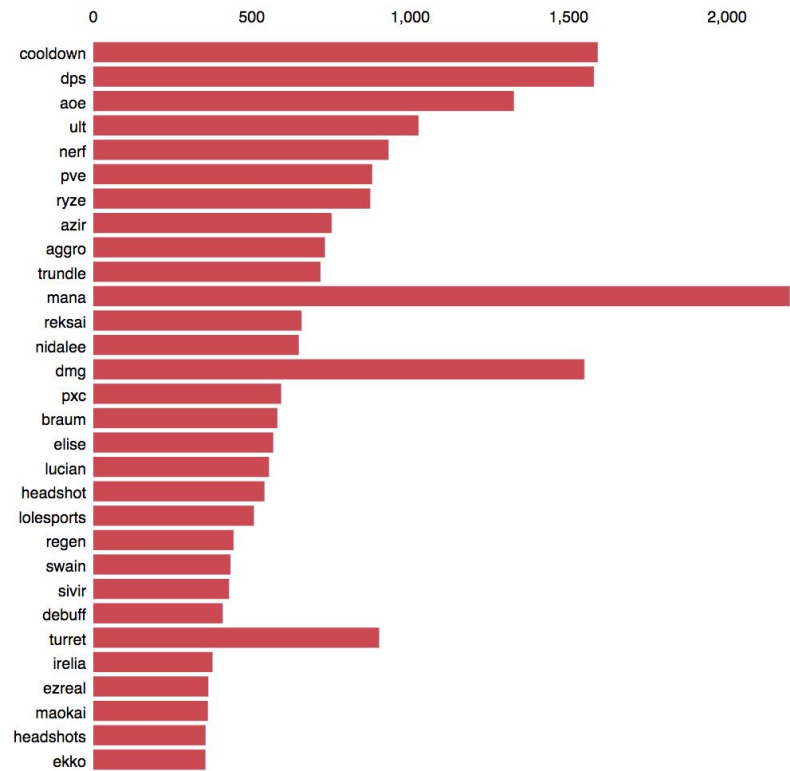
$\lambda = 0$



Intertopic Distance Map (via multidimensional scaling)



Top-30 Most Relevant Terms for Topic 4 (7.9% of tokens)



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 t; see Chuang et. al (2012)
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

PY LDA VIS

❖ Relevance Sievert & Shirley (2014)

- ※ **Lift:** ratio of a term's probability within a topic to the term's marginal probability across the whole corpus
- ※ **Lambda:** weight given to the probability of a term for a given topic in relation to its lift score

PY LDA VIS

❖ Relevance

- ※ Term w , Topic t
- ※ $\lambda * p(w|t) + (1-\lambda) * p(w|t)/p(w)$
- ※ On a log scale

PY LDA VIS

❖ Relevance

- ※ **Lambda = 1**: results ranked for terms in descending order of their topic-specific probability
- ※ **Lambda = 0**: ranks terms solely by their lift
 - Diminishes globally frequent terms out of ranking
- ※ $\lambda * p(w|t) + (1-\lambda) * p(w|t)/p(w)$

APPLICATION

- ❖ fun fact about call of duty:
- ❖ the last call of duty game you enjoyed was when the series went downhill whether that was world at war black ops mw2 black ops 2
- ❖ modern warfare feels stale today but it was incredibly fresh and new for the time
- ❖ the gameplay loop has been so refined
- ❖ enjoy also the cod vs battlefield rivalry
- ❖ arcadey shooters and neither are esports game
- ❖ sci fi games give developers more freedom in gameplay design

❖ Topic 1 : Health
※ 0.02299821

❖ Topic 2: Gaming/
Computers
※ 0.8877785

❖ Topic 7: Sports/ Gaming
※ 0.08062116

Addressable Market

*Target
Super
User*

General Audience





A Venn diagram consisting of two concentric circles. The inner circle is dark gray and labeled 'Target Super User'. The outer circle is a lighter gray and labeled 'General Audience'. The background is a textured, parchment-like surface with faint, illegible handwritten text and some geometric sketches.

Addressable Market

*Target
Super
User*

General Audience

Addressable Market

*Target
Super
User*

General Audience



Addressable Market

*Target
User*

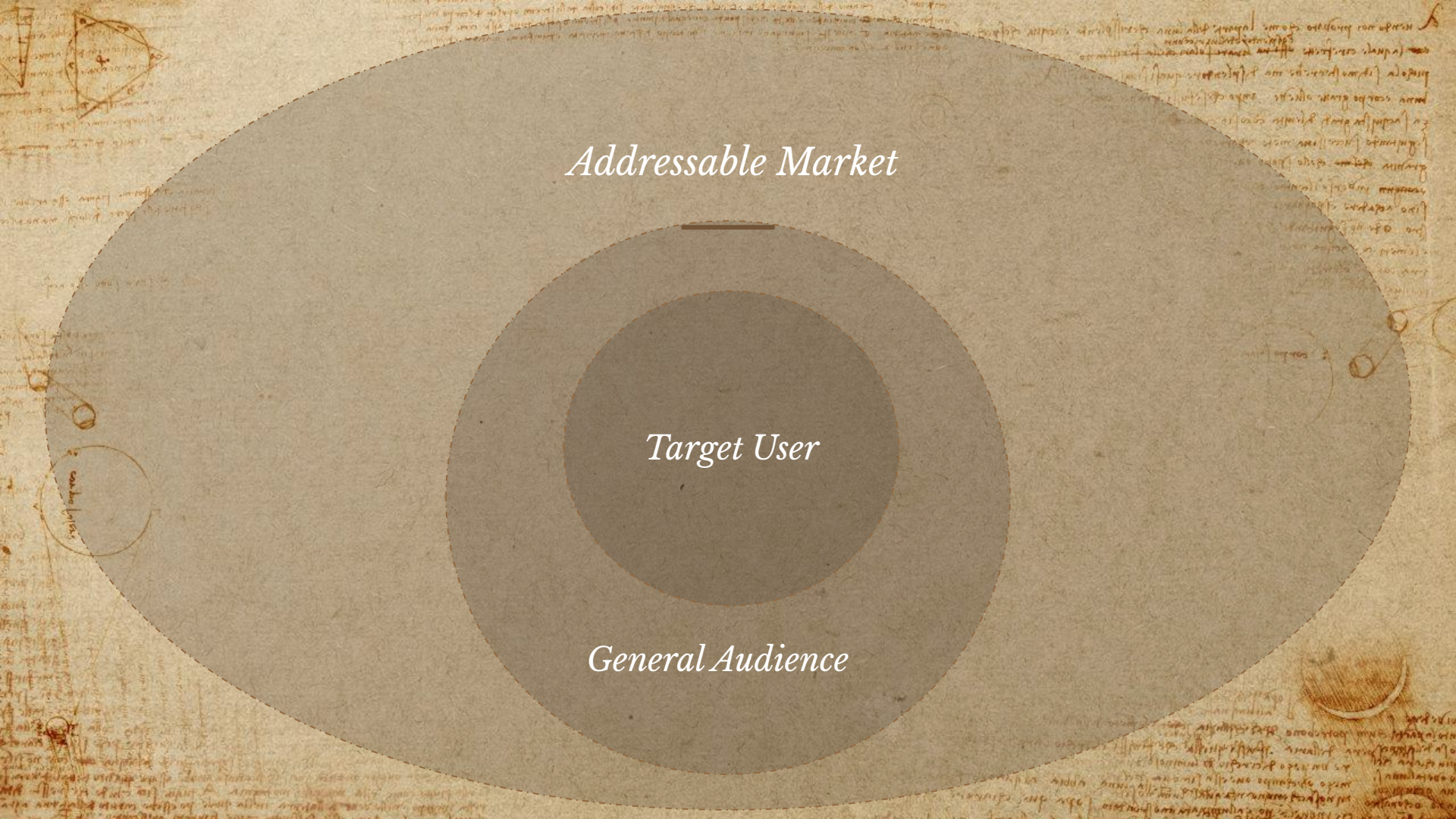
General Audience



Addressable Market

Target User

General Audience



IMPROVEMENTS

N-grams

- ❖ Improve the value of the term space
- ❖ Become more specific
- ❖ Can rethink the structure of the input and iterate the process

Word2Vec

- ❖ More opportunity to utilize similarities in the latent space
- ❖ Cut noise
- ❖ Giants' Shoulders - pretrained vectors

Non-Matrix Factorization (NMF)

- ❖ Validate LDA
- ❖ Highly interpretable and gives good results



THANK YOU

[LINKEDIN.COM/IN/DAVISVANCE/](https://www.linkedin.com/in/davisvance/)

[EMAIL: DAVISVANCE@GMAIL.COM](mailto:DAVISVANCE@GMAIL.COM)

[@DAVISDANCE](https://twitter.com/DAVISDANCE)



CREDITS

Special thanks and credits for the theme:

- ✦ Presentation template by SlidesCarnival
- ✦ Photographs by Unsplash
- ✦ Paper texture by GraphicBurger



SOURCES

- ✦ **Reddit Blog:** <https://redditblog.com/2015/12/31/reddit-in-2015/>
- ✦ **Alexa Internet:** <https://www.alexa.com/siteinfo/reddit.com>
 - ✦ Retrieved February 28, 2018
- ✦ **Wikipedia:** <https://en.wikipedia.org/wiki/Reddit>
- ✦ **Relevance:** Sievert & Shirley (2014)
<https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf>

APPENDIX

Learned Topics from Latent Dirichlet Allocation

- ❖ Spanish Stopwords (0): '0.061*"que" + 0.027*"los" + 0.017*"por" + 0.017*"del" + 0.016*"con" + 0.016*"las" + 0.014*"para" + 0.012*"una" + 0.008*"como" + 0.007*"podemos" + 0.006*"est" + 0.005*"pol" + 0.005*"pero" + 0.005*"han" + 0.004*"psoe"
- ❖ Health Products / Food (1): '0.005*"water" + 0.005*"food" + 0.004*"use" + 0.003*"make" + 0.003*"much" + 0.003*"used" + 0.003*"well" + 0.003*"skin" + 0.003*"first" + 0.003*"good" + 0.003*"eat" + 0.003*"two" + 0.002*"new" + 0.002*"ship" + 0.002*"using"
- ❖ Gaming/ Internet (2): '0.019*"game" + 0.007*"games" + 0.007*"new" + 0.007*"play" + 0.005*"think" + 0.005*"much" + 0.005*"make" + 0.004*"way" + 0.004*"see" + 0.004*"playing" + 0.004*"good" + 0.004*"want" + 0.004*"something" + 0.004*"use" + 0.004*"first"
- ❖ Empathy/ Social (3): '0.008*"want" + 0.008*"feel" + 0.006*"think" + 0.006*"going" + 0.006*"never" + 0.006*"life" + 0.005*"things" + 0.005*"much" + 0.005*"day" + 0.005*"got" + 0.005*"years" + 0.005*"said" + 0.005*"something" + 0.004*"cant" + 0.004*"friends"
- ❖ Politics/ Culture/ Global (4): '0.004*"world" + 0.003*"women" + 0.003*"men" + 0.003*"believe" + 0.003*"may" + 0.002*"war" + 0.002*"think" + 0.002*"country" + 0.002*"state" + 0.002*"government" + 0.002*"way" + 0.002*"church" + 0.002*"make" + 0.002*"say" + 0.002*"right"

APPENDIX

Learned Topics from Latent Dirichlet Allocation

- ❖ **Social/ Storytelling (5):** '0.011*"back" + 0.006*"around" + 0.006*"said" + 0.006*"got" + 0.005*"room" + 0.004*"went" + 0.004*"night" + 0.004*"door" + 0.004*"see" + 0.004*"little" + 0.004*"right" + 0.004*"still" + 0.004*"head" + 0.004*"started" + 0.004*"going"
- ❖ **Books/ Music (6):** '0.013*"chapter" + 0.012*"world" + 0.008*"song" + 0.006*"music" + 0.005*"table" + 0.005*"album" + 0.005*"links" + 0.005*"previous" + 0.005*"synopsis" + 0.005*"life" + 0.005*"source" + 0.005*"songs" + 0.004*"contents" + 0.004*"new" + 0.004*"raw"
- ❖ **League of Legends/ Gaming/ Sports (7):** '0.032*"team" + 0.017*"game" + 0.013*"match" + 0.011*"players" + 0.009*"win" + 0.009*"teams" + 0.008*"games" + 0.007*"season" + 0.007*"round" + 0.007*"play" + 0.007*"league" + 0.006*"player" + 0.006*"2016" + 0.006*"week" + 0.006*"tournament"
- ❖ **Career (8):** '0.008*"work" + 0.007*"money" + 0.006*"year" + 0.006*"job" + 0.005*"years" + 0.004*"need" + 0.004*"pay" + 0.004*"new" + 0.004*"back" + 0.004*"company" + 0.004*"going" + 0.004*"day" + 0.003*"take" + 0.003*"car" + 0.003*"help"
- ❖ **Biblical/ War (9):** '0.004*"man" + 0.004*"death" + 0.004*"lord" + 0.003*"back" + 0.003*"first" + 0.003*"see" + 0.003*"king" + 0.003*"think" + 0.003*"dead" + 0.003*"world" + 0.003*"way" + 0.003*"two" + 0.003*"well" + 0.003*"battle" + 0.003*"men"

APPENDIX

Learned Topics from Latent Dirichlet Allocation

- ❖ Entertainment (10): '0.010*"episode" + 0.009*"think" + 0.009*"show" + 0.006*"season" + 0.006*"character" + 0.005*"trump" + 0.005*"see" + 0.005*"vote" + 0.005*"story" + 0.005*"characters" + 0.005*"first" + 0.004*"series" + 0.004*"book" + 0.004*"movie" + 0.004*"good"
- ❖ Trading (11): '0.014*"card" + 0.010*"price" + 0.009*"windows" + 0.008*"purchase" + 0.007*"amazon" + 0.006*"pokemon" + 0.006*"buy" + 0.005*"shipping" + 0.005*"box" + 0.005*"key" + 0.005*"sale" + 0.005*"gift" + 0.004*"item" + 0.004*"case" + 0.004*"pro"
- ❖ Sports (12): '0.011*"game" + 0.008*"ball" + 0.008*"scores" + 0.008*"score" + 0.007*"icon" + 0.006*"line" + 0.006*"bar" + 0.005*"field" + 0.005*"2016" + 0.005*"left" + 0.005*"center" + 0.005*"right" + 0.004*"first" + 0.004*"fielder" + 0.004*"year"
- ❖ Reddit-Meta/ Community (13): '0.015*"post" + 0.011*"thread" + 0.011*"please" + 0.008*"new" + 0.008*"link" + 0.008*"questions" + 0.006*"want" + 0.006*"help" + 0.006*"reddit" + 0.005*"free" + 0.005*"community" + 0.005*"edit" + 0.005*"week" + 0.005*"comments" + 0.005*"posts"
- ❖ Gaming (Card/ MMORPG) (14): '0.011*"damage" + 0.005*"level" + 0.005*"use" + 0.004*"game" + 0.004*"attack" + 0.004*"good" + 0.004*"deck" + 0.003*"hit" + 0.003*"weapon" + 0.003*"enemy" + 0.003*"make" + 0.003*"hero" + 0.003*"kill" + 0.003*"skill" + 0.003*"speed"