CS 621 - Paper Presentation

Checking App Behavior Against App Descriptions

Ziqi Chen
Rajarshi Das
Zixin Kong

Main Contribution

How do we know a program does what it claims to do?

Research Impact:

- Computer Systems
- Interesting and novel use of Natural Language Processing.
- Usable Privacy

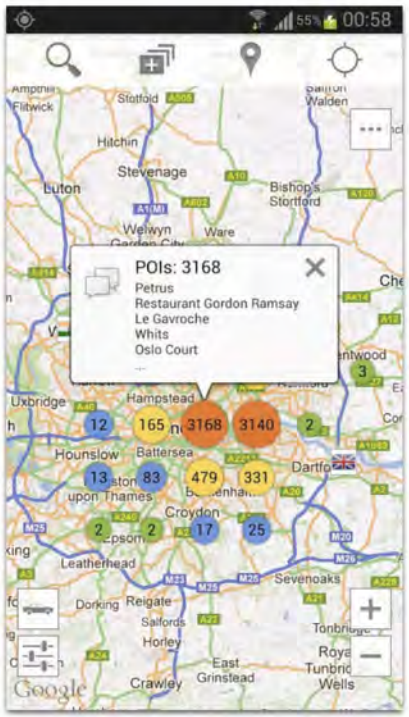
The Usable Privacy Policy Project

Towards Effective Web Privacy Notice and Choice

Natural language privacy policies have become the de facto standard to address expectations of "notice and choice" on the Web. However, users generally do not read these policies and those who do struggle to understand them. Initiatives, such as P3P and Do Not Track aimed to address this problem by developing machine-readable formats to convey a website's data practices. However, many website operators are reluctant to embrace such approaches.

https://www.usableprivacy.org/

Success story



Looking for a restaurant, a bar, a pub or just to have fun in London? Search no more! This application has all the information you need:
* You can search for every type of food you want: french, british, chinese, indian etc.
* You can use it if you are in a car, on a bicycle or walking
* You can view all objectives on the map
* You can search objectives
* You can view objectives near you
* You can view directions (visual route, distance and duration)
* You can use it with Street View
* You can use it with Navigation

Keywords: london, restaurants, bars, pubs, food, breakfast, lunch, dinner, meal, eat, supper, street view, navigation

INTERNET
GET-ACCOUNTS
ACCESS-WIFI-STATE
ACCESS-NETWORK-STATE
ACCESS-FINE-LOCATION
READ-PHONE-STATE
VIBRATE

Figure 6: The app *London Restaurants Bars & Pubs +*, together with complete description and API groups accessed

Topic Modeling in 2 mins

Probabilistic Topic Modeling

- ORGANIZE
- VISUALIZE
- SUMMARIZE
- SEARCH
- PREDICT
- UNDERSTAND

Input: An unorganized collection of documents
Output: An organized collection, and a description of how

Next few slides from Prof. Blei's talk slides
http://www.cs.columbia.edu/~blei/talks/Blei_Topic_Modeling_Workshop_2013.pdf
http://www.cs.columbia.edu/~blei/talks/Blei_Science_2008.pdf

Poisoning by ice-cream.

No chemist certainly would suppose that the same poison exists in all samples of ice-cream which have produced untoward symptoms in man. Mineral poisons, copper, lead, arsenic, and mercury, have all been found in ice cream. In some instances these have been used with criminal intent. In other cases their presence has been accidental. Likewise, that vanilla is sometimes the bearer, at least, of the poison, is well known to all chemists. Dr. Bartley's idea that the poisonous properties of the cream which he examined were due to putrid gelatine is certainly a rational theory. The poisonous principle might in this case arise from the decomposition of the gelatine; or with the gelatine there may be introduced into the milk a ferment, by the growth of which a poison is produced.

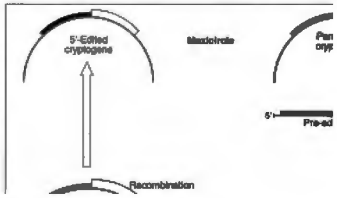
But in the cream which I examined, none of the above sources of the poisoning existed. There were no mineral poisons present. No gelatine of any kind had been used in making the cream. The vanilla used was shown to be not poisonous. This showing was made, not by a chemical analysis, which might not have been conclusive, but Mr. Novie and I drank of the vanilla extract which was used, and no ill results followed. Still, from this cream we isolated the same poison which I had before found in poisonous cheese (*Zeitschrift für physiologische chemie*, x,

RNA Editing and the Evolution of Parasites

Larry Simpson and Dmitri A. Maslov

The kinetoplastid flagellates, together with their sister group of euglenoids, represent the earliest extant lineage of eukaryotic organisms containing mitochondria (*1*). Within the kinetoplastids, there are two major groups, the poorly studied bodonids-cryptobiids, which consist of both free-living and parasitic cells, and the better known trypanosomatids, which are obligate parasites (*2*).

Perhaps because of the antiquity of the trypanosomatid lineage, these cells possess several unique genetic features (see accompanying Perspective by Nilsen)—one of which is RNA editing of mitochondrial transcripts. This RNA editing function (*3*–*7*) creates open reading frames in "cryptogenes" by insertion (or occasional deletion) of uridine (U) residues at a few specific sites within the coding region of an mRNA (5′-editing) or at multiple specific sites throughout the mRNA (pan-editing). The



Chaotic Beetles

Charles Godfray and Michael Hassell

Ecologists have known since the pioneering work of May in the mid-1970s (*1*) that the population dynamics of animals and plants can be exceedingly complex. This complexity arises from two sources: The tangled web of interactions that constitute any natural community provide a myriad of different pathways for species to interact, both directly and indirectly. And even in isolated populations the nonlinear feedback processes present in all natural populations can result in complex dynamic behavior. Natural populations can show persistent oscillatory dynamics and chaos, the latter characterised by extreme sensitivity to initial conditions. If such chaotic dynamics were common in nature, then this would have important ramifications for the management and conservation of natural resources. On page 389 of this issue, Costantino *et al.* (*2*) provide the most convincing evidence to date of complex dynamics and chaos in a biological population—of the flour beetle, *Tribolium castaneum* (see figure).

It has proven extremely difficult to demonstrate complex dynamics in populations in the field. By its very nature, a chaotically fluctuating population will superficially resemble a stable or cyclic population buffeted by the normal random perturbations experienced by all species. Given a long enough time series, diagnostic tools from nonlinear mathematics can be used to identify the telltale signatures of chaos. In phase space, chaotic trajectories come to lie on "strange attractors," curious geometric objects with fractal structure and hence noninteger dimension. As they move over the surface of the attractor, sets of adjacent trajectories are pulled apart, then stretched and folded, so that it becomes impossible to predict exact population densities into the future. The strength of the mixing that gives rise to the extreme sensitivity to initial conditions can be measured mathematically estimating the Liapunov exponent, which is positive for chaotic dynamics and nonpositive otherwise. There have been many attempts to estimate attractor dimension and Liapunov exponents from time series data, and some candidate chaotic population have been identified (some insects, rodents, and most convincingly, human childhood diseases), but the statistical difficulties preclude any broad generalisation (*3*).

An alternative approach is to parameterize population models with data from natural populations and then compare their predictions with the dynamics in the field. This technique has been gaining popularity in recent years, helped by statistical advances in parameter estimation. Good ex-

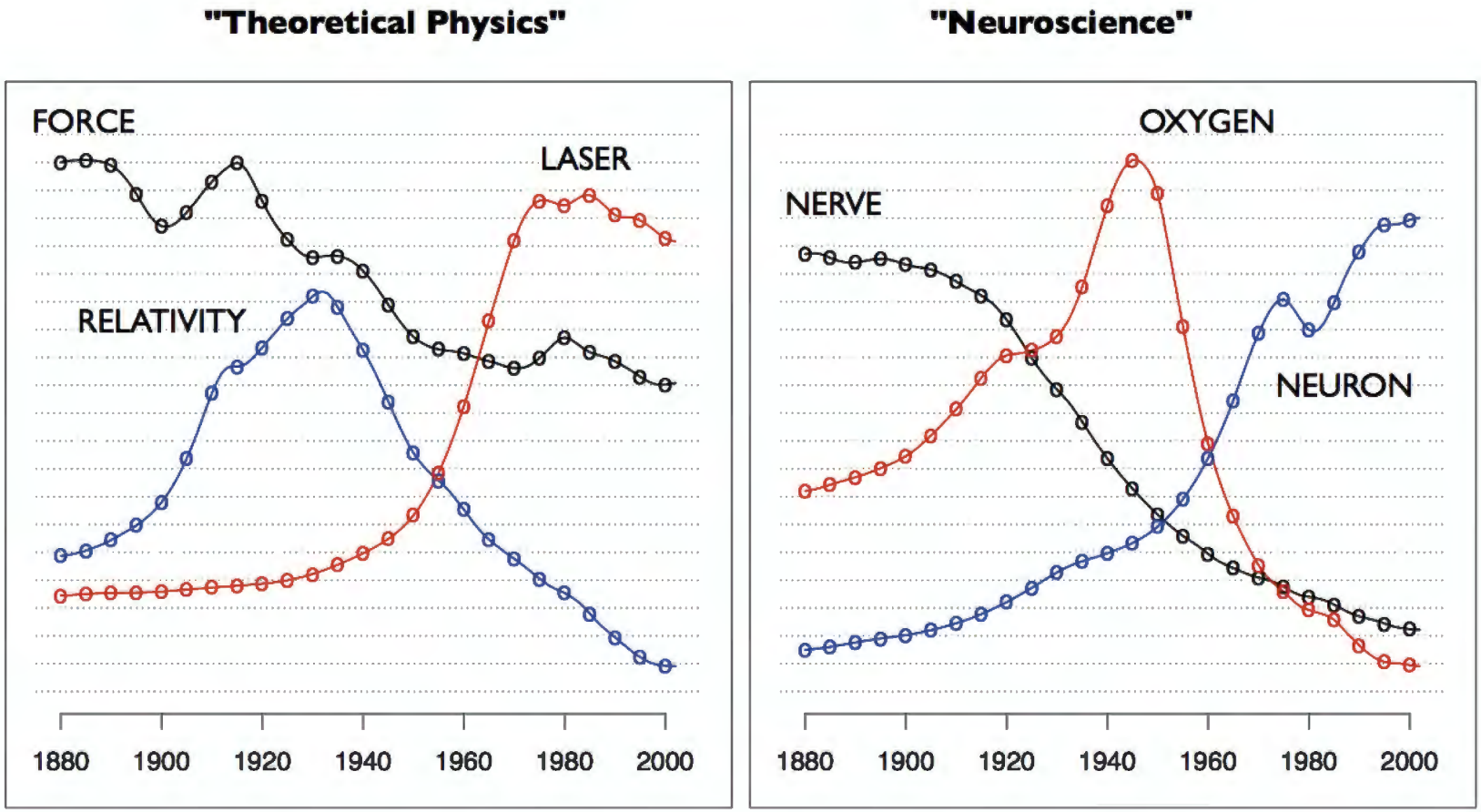
Cannibalism and chaos. The flour beetle, *Tribolium castaneum*, exhibits chaotic population dynamics when the amount of cannibalism is altered in a mathematical model

The authors are in the Department of Biology, Imperial College at Silwood Park, Ascot, Berks. SL5 7PZ UK. E-mail: m.hassell@ic.ac.uk

Topics as summary

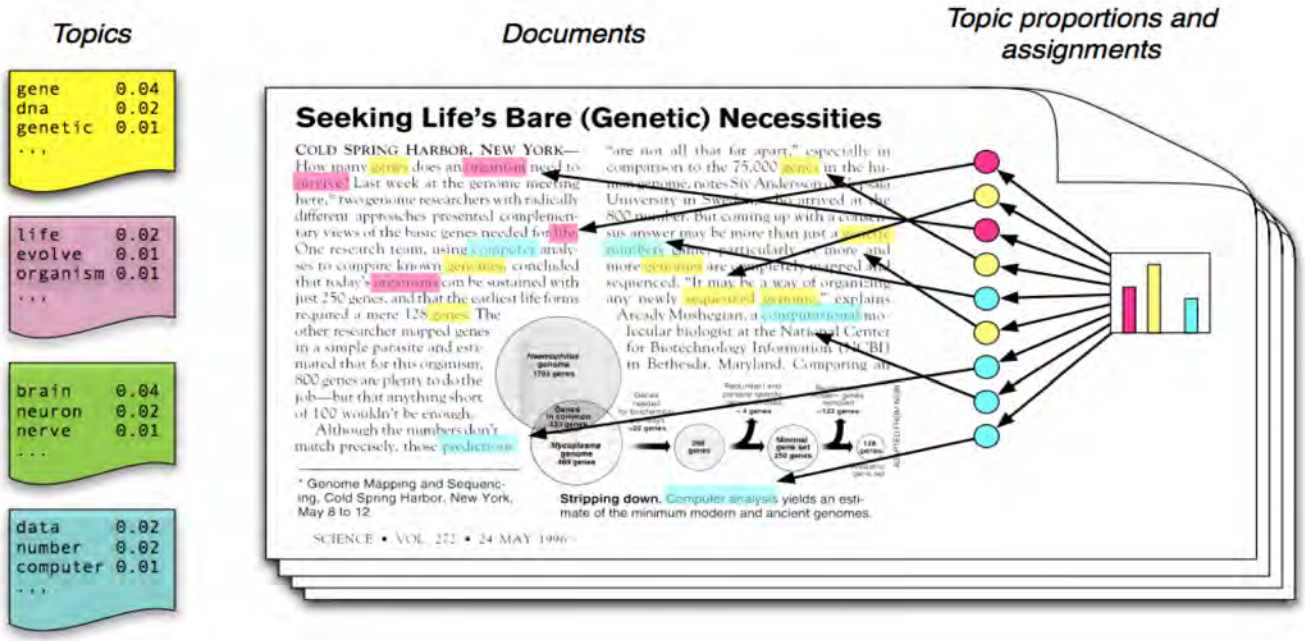
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Evolution of topics over time



Latent Dirichlet Allocation (LDA)

A document exhibits multiple topics

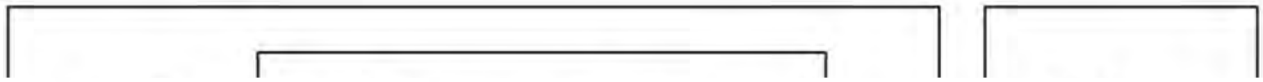


Generative process

Latent Dirichlet Allocation (LDA)

Each document is a random mixture of corpus wide topics

Each word is drawn from one of those topics



- LDA trades off two goals
  1. In each **document**, allocate its words to **few topics**.
  2. In each **topic**, assign high probability to **few terms**.

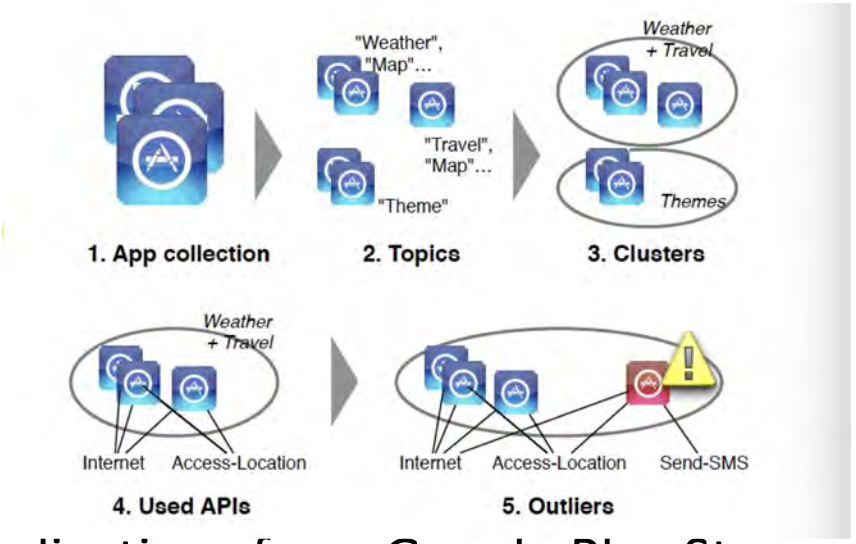
Table 1: Topics mined from Android Apps

Id	Assigned Name	Most Representative Words (stemmed)
0	"personalize"	galaxi, nexu, device, screen, effect, instal, customis
1	"game and cheat sheets"	game, video, page, cheat, link, tip, trick
2	"money"	slot, machine, money, poker, currenc, market, trade, stock, casino coin, finance
3	"tv"	tv, channel, countri, live, watch, germani, nation, bbc, newspap
4	"music"	music, song, radio, play, player, listen
5	"holidays" and religion	christmas, halloween, santa, year, holiday, islam, god
6	"navigation and travel"	map, inform, track, gps, navig, travel
7	"language"	language, word, english, learn, german, translat
8	"share"	email, ad, support, facebook, share, twitter, rate, suggest
9	"weather and stars"	weather, forecast, locate, temperatur, map, city, light
10	"files and video"	file, download, video, media, support, manage, share, view, search
11	"photo and social"	photo, friend, facebook, share, love, twitter, pictur, chat, messag, galleri, hot, send social
12	"cars"	car, race, speed, drive, vehicl, bike, track
13	"design and art"	life, peopl, natur, form, feel, learn, art, design, uniqu, effect, modern
14	"food and recipes"	recip, cake, chicken, cook, food
15	"personalize"	theme, launcher, download, install, icon, menu
16	"health"	weight, bodi, exercise, diet, workout, medic
17	"travel"	citi, guid, map, travel, flag, countri, attract
18	"kids and bodies"	kid, anim, color, girl, babi, pictur, fun, draw, design, learn
19	"ringtones and sound"	sound, rington, alarm, notif, music
20	"game"	game, plai, graphic, fun, jump, level, ball, 3d, score
21	"search and browse"	search, icon, delet, bookmark, link, homepag, shortcut, browser
22	"battle games"	story, game, monster, zombi, war, battle
23	"settings and utils"	screen, set, widget, phone, batteri
24	"sports"	team, football, leagu, player, sport, basketbal
25	"wallpapers"	wallpap, live, home, screen, background, menu
26	"connection"	device, connect, network, wifi, blootooth, internet, remot, server
27	"policies and ads"	live, ad, home, applovin, notif, data, polici, privacy, share, airpush, advertis
28	"popular media"	seri, video, film, album, movi, music, award, star, fan, show, gangnam, top, bieber
29	"puzzle and card games"	game, plai, level, puzzl, player, score, challeng, card

Table 3: Clusters of applications. "Size" is the number of applications in the respective cluster. "Most Important Topics" list the three most prevalent topics; most important (> 10%) shown in bold. Topics less than 1% not listed.

Id	Assigned Name	Size	Most Important Topics
1	"sharing"	1,453	**share** (53%), settings and utils, navigation and travel
2	"puzzle and card games"	953	**puzzle and card games** (78%), share, game
3	"memory puzzles"	1,069	**puzzle and card games** (40%), game (12%), share
4	"music"	714	**music** (58%), share, settings and utils
5	"music videos"	773	**popular media** (44%), **holidays and religion** (20%), share
6	"religious wallpapers"	367	**holidays and religion** (56%), design and art, wallpapers
7	"language"	602	**language** (67%), share, settings and utils
8	"cheat sheets"	785	**game and cheat sheets** (76%), share, popular media
9	"utils"	1,300	**settings and utils** (62%), share, connection
10	"sports game"	1,306	**game** (63%), battle games, puzzle and card games
11	"battle games"	953	**battle games** (60%), **game** (11%), design and art
12	"navigation and travel"	1,273	**navigation and travel** (64%), share, travel
13	"money"	589	**money** (57%), puzzle and card games, settings and utils
14	"kids"	1,001	**kids and bodies** (62%), share, puzzle and card games
15	"personalize"	304	**personalize** (71%), **wallpapers** (15%), settings and utils
16	"connection"	823	**connection** (63%), settings and utils, share
17	"health"	669	**health** (63%), design and art, share
18	"weather"	282	**weather and stars** (61%), **settings and utils** (11%), navigation and travel
19	"sports"	580	**sports** (62%), share, popular media
20	"files and videos"	679	**files and videos** (63%), share, settings and utils
21	"search and browse"	363	**search and browse** (64%), game, puzzle and card games
22	"advertisements"	380	**policies and ads** (97%)
23	"design and art"	978	**design and art** (48%), share, game
24	"car games"	449	**cars** (51%), game, puzzle and card games
25	"tv live"	500	**tv** (57%), share, navigation and travel
26	"adult photo"	828	**photo and social** (59%), share, settings and utils
27	"adult wallpapers"	543	**wallpapers** (51%), share, kids

Implementation



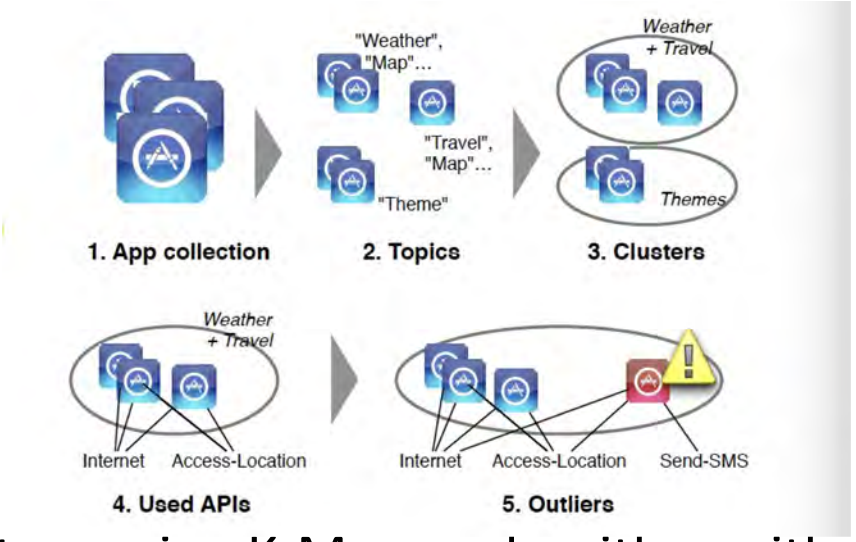
- **App collection**

CHABADA collect 22,500+ Android applications from Google Play Store

- **Identify Topics**

Using Latent Dirichlet Allocation (LDA) on the app descriptions to define 30 topics (Eg: music, money…)

Implementation



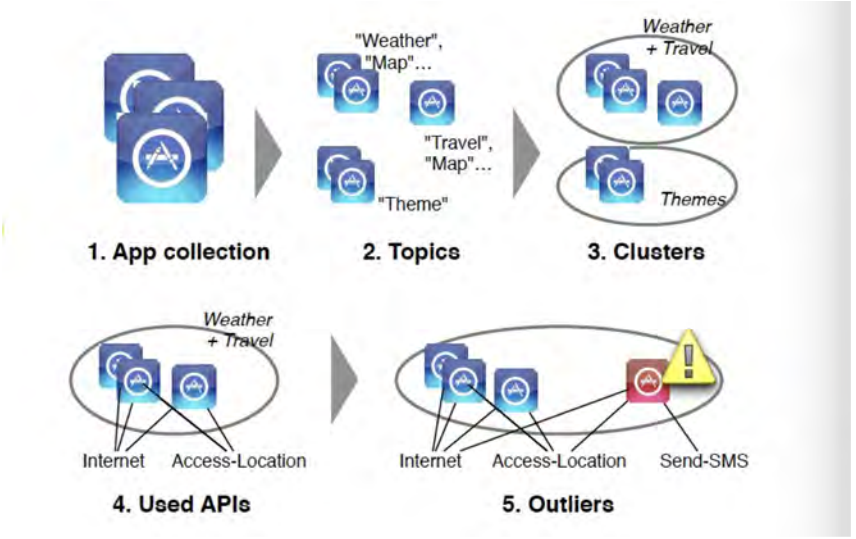
- **Cluster Apps**

CHABADA classifies apps into 32 clusters using K-Means algorithm with its related topics

- **Used APIs**

In each cluster, CHABADA identifies sensitive APIs each app statically accesses

Implementation



- **Outliers**

CHABADA identifies outliers in the APIs clusters using unsupervised one-class SVM anomaly classification

Evaluation

- **Outlier Detection**

RQ1: Can our technique effectively identify anomalies (i.e., mismatches between description and behavior) in Android applications?

- **Malware Detection**

RQ2: Can our technique be used to identify malicious Android applications?

Evaluation – Outlier Detection

- **Experiment data**

- 32 clusters with an entire set of 22,521 applications

- **Experiment - CHABADA**

- Partitioning and training: 9 subsets for training and 1 subset for testing; run 10 times

- Manual assessment: 3 categories (malicious, dubious, benign)

Evaluation – Outlier Detection

- **Result**:

- Top 5 outliers are identified from each cluster; 160 outliers out of 22,521 applications
- Top outliers, as produced by CHABADA, contain **26% malware**; additional **13% dubious** apps
- **39%** of the top 5 outliers require additional scrutiny by app store managers or end users

Evaluation – Malware Detection

- **Experiment data**
- Original set of “benign” apps (2,238) and reduced set of “malicious” apps (172)
- **Experiment:**
- Run OC-SVM as a classifier that decides whether an element would be part of the same distribution or not
- **Comparison:**
- Classification using topic clusters
- Classification without clustering
- Classification using given categories

Evaluation – Malware Detection

- **Classification using topic clusters**

	Predicted as malicious	Predicted as benign
Malicious apps	96.5 (56%)	75.5 (44%)
Benign apps	353.9 (16%)	1,884.4 (84%)

Table 1: Checking APIs and descriptions within topic clusters (CHABADA)

In our sample, even without knowing existing malware patterns, CHABADA detects the majority of malware as such.

Evaluation – Malware Detection

- **Classification without clustering**

	Predicted as malicious	Predicted as benign
Malicious apps	41 (24%)	131 (76%)
Benign apps	334.9 (15%)	1,903.1 (85%)

Table 2: Checking APIs and descriptions in one single cluster

Classifying without clustering yields more false negatives.

Evaluation – Malware Detection

- **Classification using given categories**

	Predicted as malicious	Predicted as benign
Malicious apps	81.6 (47%)	90.4 (53%)
Benign apps	356.9 (16%)	1,881.1 (84%)

Table 3: Checking APIs and descriptions within Google Play Store categories

Clustering by description topics is superior to clustering by given categories

Classification using topic clusters		
	Predicted as malicious	Predicted as benign
Malicious apps	96.5 (56%)	75.5 (44%)
Benign apps	353.9 (16%)	1,884.4 (84%)

Classification without clustering		
	Predicted as malicious	Predicted as benign
Malicious apps	41 (24%)	131 (76%)
Benign apps	334.9 (15%)	1,903.1 (85%)

Classification using given categories		
	Predicted as malicious	Predicted as benign
Malicious apps	81.6 (47%)	90.4 (53%)
Benign apps	356.9 (16%)	1,881.1 (84%)

Discussion

1. LDA is definitely not the best way to do this.

LDA only assigns probability mass to very few words in a topic and there is a high chance that a word representing malicious behavior will be left out.

Discussion

2. Problem with source of data

Discussion

3. Anomalous behavior definition not robust

Sending text messages to alert users of bad weather might be ok but not be present in the description

Discussion

4. Sensitivity to hyper-params

Discussion

5. What next? How do you plan to let the consumer's know?

Thank you!