Topick: Accurate Topic Distillation for User Streams

Anton Dimitrov*, Alexandra Olteanu*, Luke McDowell†, Karl Aberer*
*School of Computer and Communication Science
Ecole Polytechnique Federale de Lausanne
† Department of Computer Science
U.S. Naval Academy
{anton.dimitrov, alexandra.olteanu}@epfl.ch, lmcdowel@usna.edu, karl.aberer@epfl.ch

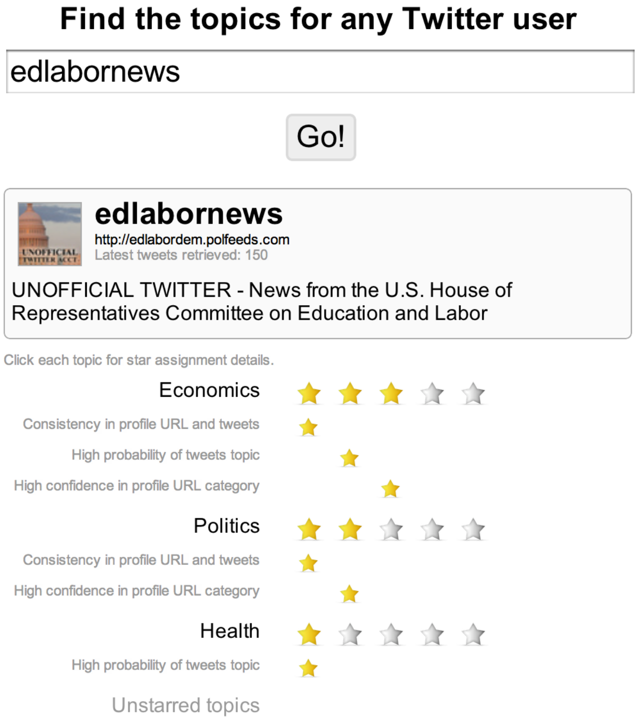
***Abstract*—Users of today's information networks need to digest large amounts of data. Therefore, tools that ease the task of filtering the relevant content are becoming necessary. One way to achieve this is to identify the users who generate content in a certain topic of interest. However, due to the diversity and ambiguity of the shared information, assigning users to topics in an automatic fashion is challenging. In this demo, we present *Topick*, a system that leverages state of the art techniques and tools to automatically distill high-level topics for a given user. *Topick* exploits both the user stream and her profile information to accurately identify the most relevant topics. The results are synthesised as a set of stars associated to each topic, designed to give an intuition about the topics encompassed in the user streams and the confidence in the results. Our prototype achieves a precision of 70% or more, with a recall of 60%, relative to manual labeling. *Topick* is available at http://topick.alexandra.olteanu.eu**

***Keywords*-Information networks; User classification; Topic models; Profile data; Twitter;**

I. Introduction

Today's online information networks[1] encourage their users to produce and disseminate information at a fast paced rate [2]. As a result, hundreds of millions of messages reach their users every day[2]. This flow of data is hard to manage and absorb, making the mechanisms for helping users to handle the information overload a necessity. In this context, a viable way to control the large volumes of data is to restrict the links to the more relevant users that produce content of interest.

In information networks, users choose what information to receive by linking to the users who produce it. As such, the links between users in information networks are directed information distribution links, rather than symmetric social links to friends or acquaintances. However, although a user A, interested in topic T, links to user B knowing that B is an expert in T, user A has no guarantee that B will actually provide information about T. In this regard, the analysis of the production and consumption of information inside information networks in conjunction with users profile information provides important clues about the topics that users are interested in, and talk about.


Find the topics for any Twitter user

edlabornews
http://edlabordem.polfeeds.com
Latest tweets retrieved: 150

UNOFFICIAL TWITTER - News from the U.S. House of Representatives Committee on Education and Labor

Click each topic for star assignment details.

Economics ★★★☆☆
Consistency in profile URL and tweets ★
High probability of tweets topic ★
High confidence in profile URL category ★

Politics ★★☆☆☆
Consistency in profile URL and tweets ★
High confidence in profile URL category ★

Health ★☆☆☆☆
High probability of tweets topic ★

Unstarred topics

Figure 1. *Topick*'s Web Interface

Let's consider the following scenario as a motivating example: a Twitter user A is interested in technology. As such, he thinks about linking to Bill Gates[3], the former chief executive of Microsoft, in order to receive the latest news about Microsoft. However, Bill is mostly generating content about topics more related with his foundation than with Microsoft. As such, user A would benefit from a service that could indicate how much Bill Gates actually talks about technology.

We introduce *Topick* a system that enables real time discovery of users' topics of interest, i.e., the topics that users discuss about inside an information network[3]. *Topick* returns for each user a list of topics along with a number of stars that are meant to reflect how much the user actually talks about those topics. To do so, *Topick* performs a

[1] E.g.,Twitter, Flickr, Digg, YouTube

[2] http://blog.twitter.com/2011/06/200-million-tweets-per-day.html

[3] http://en.wikipedia.org/wiki/Bill_Gates

thorough analysis (a) of the content produced by the user, and (b) of what the user declares about herself. To demonstrate *Topick*'s functionality, Figure 1, we implemented it for Twitter[4], a popular information network. However, the processing framework behind *Topick* can be adapted to other information networks as well.

The main two observations behind *Topick* are: (1) the language used by *subject-matter experts* in a given topic and their entourage is *representative* for the way in which users discuss about the topic inside a specific information network; and (2) many users provide rich information about themselves, i.e., profile information, in the form of URLs[5] that point to, for instance, personal/business websites. In contrast with previous works [4], [6], [5], we leverage both the content produced by users inside information networks (e.g., posts, tweets) and the personal information that they provide about themselves, which we refer to as *profile data*. Thus, the topic discovery algorithm behind *Topick* inspects if the user stream contains similar vocabulary as the one used by subject-matter experts in a certain topic, and categorizes the website to which her *profile* points to with an external text classification service[6].

Prior work addresses topic discovery in information networks: TweetMotif [4] is a Twitter search application that groups the search results in a set of different subtopics to facilitate navigation. Their subtopics represent different contexts in which the search query is discussed. TwitterRank [6] uses topic modeling to extract users topics of interest with the purpose of understanding how reciprocity correlates with the similarity between topics of interest. Much closer to our work, Twopics [3] and TweetLDA [5] try to infer user's topics of interest. Twopics's approach relies on entities discovery, which are then mapped to high-level topics[3]. In contrast, we use topic modeling, i.e., Latent Dirichlet Allocation (LDA)[1], to distill topics from user content, and AlchemyAPI to categorize her profile data. TweetLDA [5] employs topic modeling to assign topics to Twitter users, yet, they only evaluate the relative performance of such an approach compared to external text classification services.

II. System Overview

Topick automatically detects Twitter users' interests in a set of predefined high-level topics. Once *Topick* receives the user data, it performs three main processing steps. First it cleans the user content and distils the topics using LDA[1] and a pre-computed topic model. Then, the AlchemyAPI[6] is used to classify the website pointed by the user profile URL. Finally, we combine the obtained results to assign users a number of stars for each topic, which reflects their interest in the topic. Next, we detail each of these steps.

A. Topic Detection

Data cleaning Being short messages – limited to 140 characters, the user tweets are poor in information, noisy (e.g., use of acronyms, words containing symbols, words mixture) and of varying quality. Therefore, we concatenate a user tweets in a single *document* [6], which we then clean before processing: first short and noisy words are removed (i.e., words with less than 3 letters and/or containing non-English symbols); then, we tokenize and stem the remaining words; next, we filter out a list of predefined stop words[7]; finally, we also remove too frequent or too sporadic words.

Topic Distillation To distil the *topics* in a user tweets, *Topick* employs LDA, a probabilistic topic model that assumes that each *document* (i.e., seen as a "bag of words") exhibits multiple topics, yet ignores the correlations between topics. First, LDA needs to learn a topic model with a given number of topics T from a set of training documents. Here, a topic is a probability distribution over the *vocabulary* of the training documents. Then, using the obtained topic model, LDA represents each document as a probability distribution over a set of topics.

When classifying new users, LDA computes the probability distribution over the obtained topics only. As a result, a document can get a high probability for a topic T without necessarily discussing about T – for instance, it contains many new words and only a few words that have high probability to belong to topic T. To avoid incorrect assignment of topics in such cases, we define a simple score that uses the LDA topic model (i.e., the probability of each word to belong to a topic): for each document and each topic we compute the weighted average of the words probabilities; the probabilities are weighted by the minimum between the word frequency and a maximum allowed weight (to avoid weighting a certain word too much). Though simple, our results (Section II-C) show that this score preserves a high precision while improving the recall.

Profile Data Classification To categorize the URLs provided by users, *Topick* relies on AlchemyAPI, which classifies each webpage into some predefined categories[8]. Along with the returned category, AlchemyAPI also provides a confidence score as a value between 0 and 1 (higher is better).

B. Star System

To sum up, three different *base scores* can be computed for each user in real time – two content-based and one profile-based: (a) S_{LDA} – representing the LDA probability that a user discusses a certain topic; (b) S_{avg} – the weighted

[4] https://twitter.com/about

[5] In our dataset 80% of the users provide URLs

[6] AlchemyAPI: http://www.alchemyapi.com/api/

[7] From: http://dev.mysql.com/doc/refman//5.5/en/fulltext-stopwords.html, and http://www.lextek.com/manuals/onix/stopwords1.html

[8] Arts & Entertainment, Business, Computers & Internet, Culture & Politics, Gaming, Health, Law & Crime, Religion, Recreation, Science & Technology, Sports and Weather; Details at http://www.alchemyapi.com/api/categ/categs.html

Health	Sports	Economics	Art	Fashion	Politics	Technology
health 0.0145	game 0.0136	bank 0.0066	feel 0.0064	photo 0.0180	obama 0.0082	googl 0.0081
news 0.0066	team 0.0087	econom 0.0046	well 0.0052	fashion 0.0079	romney 0.0056	social 0.0081
care 0.0057	play 0.0079	market 0.0045	wait 0.0048	design 0.0072	video 0.0050	post 0.0078
studi 0.0054	race 0.0062	china 0.0041	nice 0.0044	post 0.0069	tonight 0.0047	blog 0.0076
learn 0.0051	player 0.0056	rate 0.0041	night 0.0043	open 0.0060	vote 0.0047	facebook 0.0058
cancer 0.0046	well 0.0049	report 0.0041	tweet 0.0041	video 0.0057	state 0.0040	media 0.0053
patient 0.0045	coach 0.0038	polit 0.0040	better 0.0040	beauti 0.0049	stori 0.0038	twitter 0.0050
report 0.0044	final 0.0037	govern 0.0036	play 0.0038	collect 0.0049	presid 0.0038	video 0.0048
medic 0.0043	lead 0.0037	news 0.0035	didn 0.0038	spring 0.0045	join 0.0037	free 0.0046
join 0.0042	sport 0.0036	interest 0.0032	amaz 0.0037	artist 0.0045	american 0.0036	busi 0.0044
nation 0.0041	score 0.0035	price 0.0031	friend 0.0036	featur 0.0042	hous 0.0035	appl 0.0044
women 0.0040	season 0.0035	labour 0.0031	thought 0.0035	style 0.0039	news 0.0035	data 0.0042
school 0.0036	win 0.0034	polici 0.0030	pretti 0.0034	blog 0.0039	book 0.0033	market 0.0042
food 0.0033	fan 0.0033	growth 0.0030	enjoy 0.0033	galleri 0.0038	paul 0.0030	ipad 0.0038
drug 0.0033	footbal 0.0033	economi 0.0030	morn 0.0033	shop 0.0037	bill 0.0030	mobil 0.0037
support 0.0033	goal 0.0032	london 0.0028	long 0.0032	facebook 0.0036	interview 0.0029	interest 0.0035
risk 0.0031	tonight 0.0030	busi 0.0028	life 0.0032	dress 0.0036	campaign 0.0027	onlin 0.0034
student 0.0030	winner 0.0030	fund 0.0028	sound 0.0030	free 0.0035	report 0.0027	email 0.0032
children 0.0030	leagu 0.0029	vote 0.0027	hear 0.0030	exhibit 0.0035	john 0.0026	design 0.0032
heart 0.0030	hors 0.0028	financi 0.0027	girl 0.0028	photograph 0.0034	support 0.0025	digit 0.0032

Table I
TOPICS EXTRACTED BY LDA

average of the probabilities of the words belonging to the document (i.e., user tweets) to refer to a certain topic; and (c) $S_{AlchemyAPI}$ – the confidence that the category returned by AlchemyAPI is the correct one. Motivated by the fact that none of these scores achieve a good tradeoff between precision and recall by itself (see II-C), we devised a rule-based system that combines the above scores to assign stars to users for the discovered topics:

1) *High Scores* When a user obtains a high score (e.g., high probability, high confidence) for a certain topic, the topic assignment is done with higher precision. Hence, when the scores surpasses a given threshold[9] for a certain topic, a star is assigned on that topic.
2) *Topic Consistency* Inferring the same topic from different data sources (i.e., user stream and user profile data) gives more confidence that the topic assignment is correct. When either one of the highest content-based scores (i.e., S_{LDA} and S_{avg}) is obtained for the topic returned by AlchemyAPI, we assign one more star to this topic. This rule applies even when the obtained scores do not surpass the given thresholds.
3) *Confidence Interval* Obtaining similar content-based scores for different topics might indicate that either the user has an interest in more topics, or that the algorithm is not able to identify the dominant topic. In such cases, we apply the above consistency rule for all topics that obtained a score in the interval above 90% of the highest score.

Properties By design, *Topick* exhibits the following properties: (1) *explainability* – *Topick* is able to explain why a star was assign for a certain topic (e.g., based on what rule and what type of user data); (2) *confidence* - a higher number of stars assigned for a certain topic results in a higher assignment precision, and, thus, a higher confidence in the assignment; (3) *multiple topics per user* - the application does not return only the most probable topic of interest, but a combination of topics, since the user might have an interest in several topics.

C. Evaluation

Data Collection To generate a good topic model that reflects how Twitter users discuss about a set of high-level topics, LDA needs a representative set of documents (i.e., Twitter users tweets) to learn from. In this regard, we built a dataset based on the following observations: (1) people with similar background are more inclined to be connected and interact with each other [2], and (2) the language used by *subject-matter experts* and their entourage is representative for the way in which the Twitter users discuss about a certain topic. We started building the dataset from 160 manually selected *core users*, labeled as *subject-matter experts* in 7 topics: *Technology, Sports, Politics, Economics, Arts, Fashion and Health*. Then, we fetched the followees of the core users with less than 400 followees[10], which resulted in $13,573$ users. Finally, for all these users, we retrieved the most recent 150 tweets (provided that the users had at least this amount of tweets) resulting in a total of 2.5 million tweets.

Topic Extraction Since this dataset is a mixture of content produced by *subject-matter experts* in 7 topics and their entourage, we want to see if, by using LDA, we can build a topic model whose topics easily reflect those of the manually selected subject-matter experts. In Table I, we show the words having the highest probability to belong to each topic. Since the obtained topics are clear enough to be labeled with the topics of the subject-matter experts, we use the obtained model to classify the content produced by Twitter users in one of these topics.

Performance Evaluation To evaluate *Topick* we measured the *precision* – the number of *correctly* classified users

[9] The thresholds for the base scores were experimentally chosen to ensure a good tradeoff between precision and recall, and based on the assumption that only a small percentage of users from the entire population have an interest in a given topic.

[10] Decision motivated by the Twitter API's 350 requests per hour limit.

Score	Threshold	Core Users		Test Set 1		Total		
		Prec.	Recall	Prec.	Recall	Prec.	Recall	F1
S_{LDA}	0.72	0.89	0.21	1	0.17	0.93	0.19	0.31
S_{avg}	$0.5 * 10^{-3}$	0.87	0.37	0.87	0.27	0.87	0.33	0.47
$S_{AlchemyAPI}$	0.84	0.82	0.17	0.96	0.18	0.88	0.18	0.3
No. of Stars	1	**0.73**	**0.61**	**0.68**	**0.58**	**0.70**	**0.6**	**0.64**
No. of Stars	2	0.83	0.49	0.74	0.47	0.79	0.48	0.59
No. of Stars	3	0.92	0.3	0.90	0.35	0.91	0.32	0.47
No. of Stars	4	1.00	0.14	1.00	0.13	1.00	0.14	0.24
No. of Stars	5	1.00	0.05	1.00	0.016	1.00	0.035	0.06

Table II
PERFORMANCE EVALUATION IN TERMS OF PRECISION, COVERAGE, ACCURACY.

Test Set 2			Total		
Prec.	Recall	F1	Prec.	Recall	F1
0.82	**0.68**	**0.74**	0.70	0.6	0.64

Table III
PERFORMANCE EVALUATION IN TERMS OF PRECISION, COVERAGE, ACCURACY.

divided by the number of classified users and the *recall* – the number of correctly classified users divided by the total number of users, which we use to compute the *F1 score*. Then, to measure *Topick*'s relative performance with respect to the human labeling ability, we used 3 different test sets: (a) the *core users* – containing the initial 160 manually labeled users as *subject-matter experts*, (b) *test set 1* – containing 125 randomly selected and manually labeled users from our dataset, and (c) *test set 2* – 100 users selected using the Twitter's Browse Categories functionality[11] that suggests people to follow if interested in a given topic. While the core users set and the test set 1 belong to the dataset used by LDA to learn the topic model, test set 2 was build independently from this dataset, and we used it to analyze if our model generalizes well.

Table II summarizes the results for the star system behind *Topick*, and for each base score individually on the core users set and the test set 1. Among the base scores, S_{avg} preforms the best, obtaining the highest F1-score. Combining the three base scores in the *Topick*'s star system leads to better results than when used apart, resulting in a F1-score of 0.64. Moreover, it can be seen that the higher the threshold for the star score is, the higher the precision. Reaching perfect precision for 4 stars comes at the cost of low recall. Next, we looked to see if our model has general applicability as well: Table III compares the results obtained on the core users set and the test set 1 together, with those obtained on test set 2[12]. The slightly better results obtained on the test set 2 might be explained by the quality of Twitter suggestions. We note that our model, generalizes well and successfully classifies new users. Finally, one should keep in mind that human labeling is susceptible to introduce biases, thus not perfect. As such, we consider the obtained results reasonable.

III. PROTOTYPE AND DEMONSTRATION SCENARIO

We built *Topick* as a Python-based web application, to demonstrate how our star score system can be used in practice to classify Twitter users in real time. The web interface (Figure 1) asks the user to introduce a Twitter screen name. In return, it displays a profile summary, along with the topic star assignment, and a brief justification of the assignment.

The star assignment is computed in real time. A *Topick* request is handled by first fetching the latest tweets from the user and invoking Alchemy API on the latest version of the profile URL. Then, the LDA algorithm is used to compute the probability distribution over the topics. The three base scores are then computed, and the star rules are applied to generate the final set of topics and their number of stars, which are displayed back as a result on the web page.

The user can click on each topic assignment to reveal a justification for each assigned star (Figure 1 shows all the topic justifications revealed). The justification is broken down into up to three parts (corresponding to the rules in the star system): the consistency in profile URL and tweets, the probability of the tweets' topic, and the confidence in the profile URL category. The topics that were not selected by the system are shown in gray and are folded by default to avoid cluttering the user interface.

IV. CONCLUSIONS

We introduced *Topick*, a system that automatically distills high-level topics for users in information networks. Our prototype achieves a precision of 70% or more, with a recall of 60%, compared with human labeling. *Topick* is available at http://topick.alexandra.olteanu.eu.

Acknowledgements We are indebted to Stefan Bucur for his valuable feedback and help on the earlier versions of this work. This work was partially supported by the grant *Reconcile: Robust Online Credibility Evaluation of Web Content* from Switzerland through the Swiss Contribution to the enlarged European Union.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 2003.

[2] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10*.

[3] Matthew Michelson and Sofus A. Macskassy. Discovering users' topics of interest on twitter: a first look. In *AND'10*.

[4] Brendan O'Connor, Michel Krieger, and David Ahn. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In *ICWSM'10*.

[5] Daniele Quercia, Harry Askham, and Jon Crowcroft. TweetLDA: Supervised topic classification and link prediction in twitter". In *WebSci'12*.

[6] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM'10*.

[11] https://twitter.com/who_to_follow/interests

[12] The threshold for the star score is set to 1 star