A Network Tour of NIPS Conference Papers

Group 26: Yueran Liang, Peilin Kang, Yawen Hou, Zhechen Su
*EPFL - EE-558 (*A Network Tour of Data Science*)*

I. Introduction

Machine learning has always been a very popular subject since the last decade. In the last few years, the number of papers submitted to the top machine learning conferences has been increasing exponentially. More and more researchers become interested to this field and new concepts and algorithms are proposed every year at conferences. Among all the conferences, Neural Information Processing Systems (NIPS) is the one of the world's largest machine learning conference (see section IV). For our current project, we will build our networks based on NIPS papers. We will study the papers' topics and their authors' favorite topics by building a network of topics using Latent Dirichlet allocation (LDA) and associate each researcher to the topics they write the most about using Author-Topic modeling (ATM). We will also study the connections between individual researchers by building a co-authorship graph using adjacent matrices. To better visualize the networks, we will apply dimensionality reduction to our network models using T-distributed Stochastic Neighbor Embedding (t-SNE). We will exploit the distribution the distribution of the papers' topics over years and the distribution of topic preference per author. As a result, we will attempt to predict the main topic of a new paper with our LDA model. Given a particular author, we will also attempt to look for other authors that have the same research orientation as them.

II. Data Preprocessing

Our dataset includes the NIPS papers and an extracted list of the authors of these papers. To be able to build our topic model from the NIPS papers, we need to process them. We extract the raw textual data of each paper, and transform them through several steps before getting the necessary data for topic modelling.

1) **Bag of words** English sentences are composed of punctuation, spaces, and words. We first need to split each sentence into a bag of words by extracting the punctuation and the spaces. For example, *Nobody knows how ancient people started using fire* will be processed to {*Nobody, knows, how, ancient, people, started, using, fire*}.
2) **Remove stop words** In English, there are a lot of words such as *a*, *the*, *or*, etc., that have a high frequency of appearance in sentences without bearing any specific meaning. These words are called stop words, and are often used as articles, prepositions, adverbs or conjunctions. We will remove all of them in order to extract the truly meaningful words that express the central ideas of the papers. Another benefit would be, by removing these words, it will also take less time to train the model and it will help in improving the accuracy of the model. Following the example that we have given in the previous section, from {*Nobody, knows, how, ancient, people, started, using, fire*}, we will remove the stop words {*knows, how, using*} to get {*Nobody, ancient, people, started, fire*}.
3) **Construct bi-grams** Some words do not mean anything alone and only make sense when they are paired with another word. For example, *York* alone means nothing, but *New York* is the name of a city. A pair of two consecutive words frequently occurring together is called a bi-gram. We will search for these bi-grams and for each pair, we will concatenate them to treat them as a new word. In order to do this, we set a low and a high threshold to filter the frequency of appearance of each pair of words. If a pair do no appear frequently enough, we ignore it because it probably means that they do not form a bi-gram. If two words appears to frequently together, we will also ignore them because we might have found words like *et al.* that do not carry any particular meaning. As an example, {*Nobody, ancient, people, started, fire*} will become {*Nobody, ancient_people, started, fire*} after this step.
4) **Stemming** English nouns have the singular and plural forms and English verbs can be conjugated in different tenses. Even though the word may be written in diverse forms, its meaning does not change. Therefore, we need to treat all possible variations of a words as one single term. For example, *apple* and *apples*, *doing* and *done* need to be treated as if they were the same word. In order to do so, we will extract the stem of these words, which is their the most basic form. On the same time, we will all the capitalization of the words.

For instance, {*Nobody, ancient_people, started, fire*} is processed into {*nobody, ancient_people, start, fire*}

III. Models

A. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for discrete data collections, such as text corpus, first proposed by Blei et al. [1]. The concept of LDA is based on the assumption that documents are generated from multiple topics. The document generation process using LDA is also based on this assumption. We first build our dictionary using the word stems that come from the data preprocessing (sec. II). Then, we will count the term frequency and the document of each word using this dictionary. At last, we will build our LDA model and extract the topics. Each topic is represented as an array of words that are closely related to the topic. For example, the topic array of "gene" will contain with high probability words like {*DNA, chromosome, cell, etc.*}.

For each document in the document collection, the LDA model (Figure 1) processes every word in the text as follows:

1) For each of the T topics, draw the word distribution ϕ independently from a symmetric Dirichlet (β) prior;
2) Randomly choose the topic distribution θ in the document from a symmetric Dirichlet (α) prior;
3) Choose a topic z responsible for generating that word, drawn from the θ distribution;
4) Choose word w from the topic distribution θ corresponding to z.

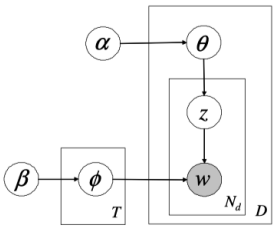

Fig. 1. Latent Dirichlet allocation (LDA)

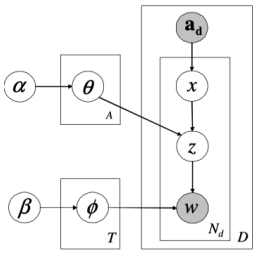

Fig. 2. Author-Topic Model (ATM)

This hierarchical Bayesian model estimates the parameters ϕ and θ which provide information about the topics that are present in a corpus and the weights of these topics in each document respectively. The core computational problem of topic modeling is the use of observed documents to infer hidden topic structures. Gibbs sampling [2] is used to estimate these parameters. This can also be seen as the inverse of the generative process of LDA.

B. Author-Topic Model

The Author-Topic Model (ATM) proposed by Rosen-Zvi et al. [3] is an extension of Latent Dirichlet Allocation (LDA) that includes authorship information. This model allows us to learn the topic representations of authors in a corpus. LDA describes each document as a mixture of probabilistic distribution of topics and each topic as a multinomial distribution over words. ATM allows to add an additional author layer by assuming that each author is associated with a multinomial distribution over topics. A document with multiple authors is represented as a mixture of the topic distributions affiliated with the authors. As a result, every author is associated with multiple documents, and each document can be associated with multiple authors.

The overall process is shown in Figure 2. Different from LDA, the ATM model's document generation process will work like this:

1) From a group of authors a_d, choose an author x associated with the generation of this word;
2) Associate author x with a distribution θ over topics, chosen from a symmetric Dirichlet (α) prior;
3) Select a topic z according to the author and their topic distribution generated in step 1 and 2;
4) For each of T topics, draw words distribution ϕ independently from a symmetric Dirichlet(β) prior;
5) Generate a word according to the distribution ϕ corresponding to topic z.

The author-topic model subsumes the topic model and author model. Indeed, the topic model corresponds to the case where each document has one unique author, and the author model corresponds to the case where each author has one unique topic. Estimating the parameters ϕ and θ through Gibbs sampling [2], we obtain information about which topics authors typically write about, as well as a representation of the content of each document in terms of these topics.

IV. Dataset

Our dataset comes from the Neural Information Processing Systems (NIPS) conference, which involves fields (topics) like machine learning, cognitive science, psychology, computer vision, statistical liguistics, and information theory. We modified the scraper from a Kaggle dataset [1] and scraped NIPS papers published from 1987 to 2018

[1] https://www.kaggle.com/benhamner/nips-papers

Nb of Authors	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	17	18	20	21	8978
Nb of Connected Components	211	291	192	95	44	26	24	7	5	2	4	4	2	2	1	1	1	1	1	1

TABLE I
OVERVIEW OF CONNECTED COMPONENTS IN THE CO-AUTHORSHIP NETWORK.

from the NIPS website [2]. There are 9719 authors who published 7241 documents in total. The scraped data is saved in 3 csv files namely Authors.csv, Papers_authors.csv and Papers.csv. Authors.csv contains each author's id associated with their full name. Papers.csv includes the paper's id and the paper's title. Papers_authors.csv connect each paper to its author(s) by ids. We will use Papers_authors.csv to generate our co-authorship network, and Papers.csv to generate our topic model.

V. CO-AUTHORSHIP NETWORK

We establish the co-authorship network (Figure 4) by connecting each pair authors together if they have written the same paper. The figure 4 shows that there exist a very large connected component in the network. There are 915 connected components in total and the detail of the connected component is shown in TableI. The first column depicts the number of authors in the connected component, and the second column indicates the existing number of connected components having this number of authors.

The largest connected component have 8978 authors. The diameter of the network in largest connected component is 5.95 with a clustering coefficient of 0.7 and a density of 0.006. We have found that a great number of researchers collaborate with each other in their redaction of the NIPS papers. After plotting the degree distribution of the network (Figure 3), we find that the co-authorship network follows the power-law distribution with large number of authors having very few degrees. This also illustrates the fact that only few researchers have a very high academic performance and have more chance to collaborate with other researchers. Another possible reason to this is that these central researchers are head of one or several laboratories, thus we can also find their names on the multiple papers written the people from their lab.

We have also calculated the betweenness centrality for each node. The average of the betweenness centrality is 0.000552 ± 0.002876 and its maximum is 0.142048. Again, most of the authors have a very small centrality, which is coherent with our findings from the degree distribution graph and the illustation of the network. Many researchers only publish a few number of papers on NIPS, therefore it is logical that they have not collaborated with many other authors. We can conclude that authors with high centrality are the big figures that have large influence in these fields. To illustrate our sayings, we have also plotted the betweenness centrality for each author (Figure

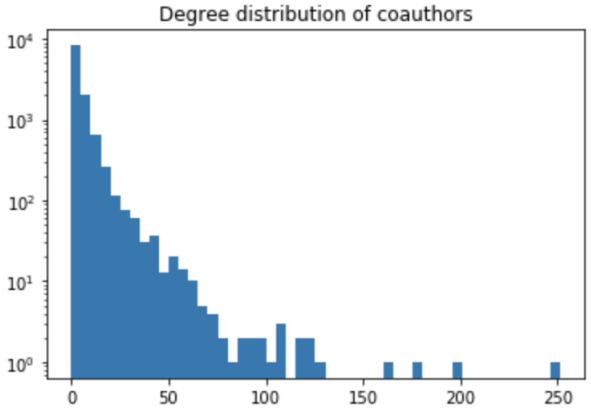

Fig. 3. Degree Distribution of Co-authorship Network

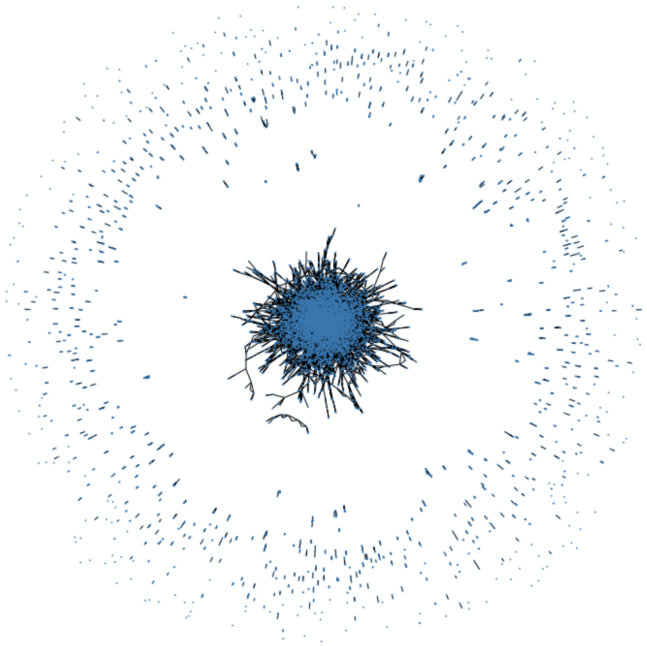

Fig. 4. Co-authorship Network. The ball-like component in the middle is the largest connected component.

[2]https://papers.nips.cc/

5). From the plot, it shows that Michael I. Jordan, Bernard Schölkopf, Zoubin Ghahramani, Yoshua Bengio and Tong Zhang are the top-5 most influential authors.

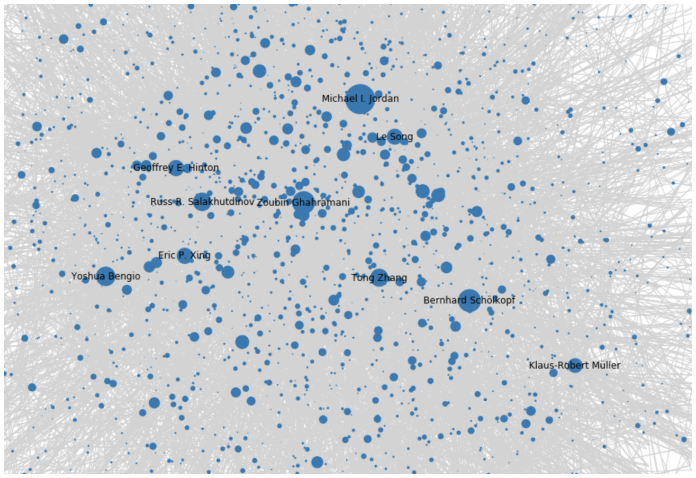

Fig. 5. Researchers having a great influence

We have also build a smaller graph connecting only authors who have collaborated together for at least 3 times (i.e. they have written 3 papers together). In this small world, the largest connected component contains 1321 authors. While exploring the betweenness centrality of this largest connected component, we have arrived to same conclusion as above.

VI. Topic Network

We decide to construct our topic network based on the co-authorship between researchers. As each researcher has their main research field, we represent them in by the topics of their research direction. We can derive the topics and the topic preference of each author using the ATM model. In our topic network, each node represent a topic (a mixture of words). Each node can be associated with a group of author whose main research direction corresponds to this topic. We connect two nodes if two authors working on these two topics have collaborated together in the redaction of a paper. The network is shown in Figure 6. It shows that authors of Guassian Processes (with keywords e.g. bayesian,latent), Bandit Algorithm (e.g. bind,lemma) and Neural Network (e.g. gradient,neural network) collaborated most frequently together.

For visualisation, we applied dimensionality reduction using t-SNE technique, first proposed by van der Maaten et al. [4], to observe the distribution of the authors' topic preference. As shown in the Figure 7, the position of the center of each circle represents the preference of each

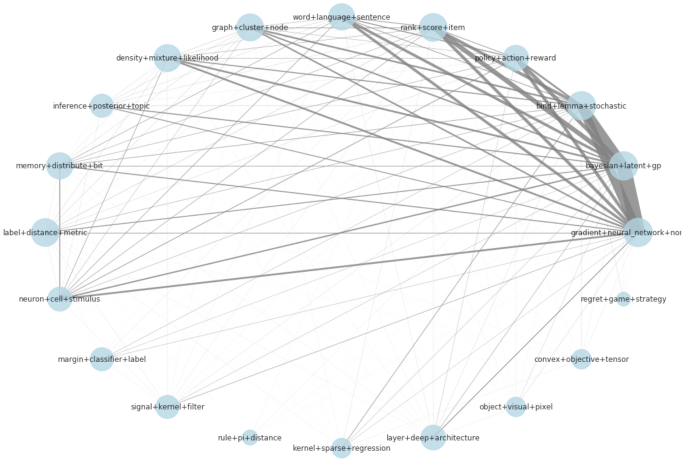

Fig. 6. Overview of the Topic network.

author. The size of the circle represents the number of the papers this author has written. In order words, the larger the circle, the more papers they write. If the centers of two circles are very close to each other, it means that the topic preferences of these two authors are very similar.

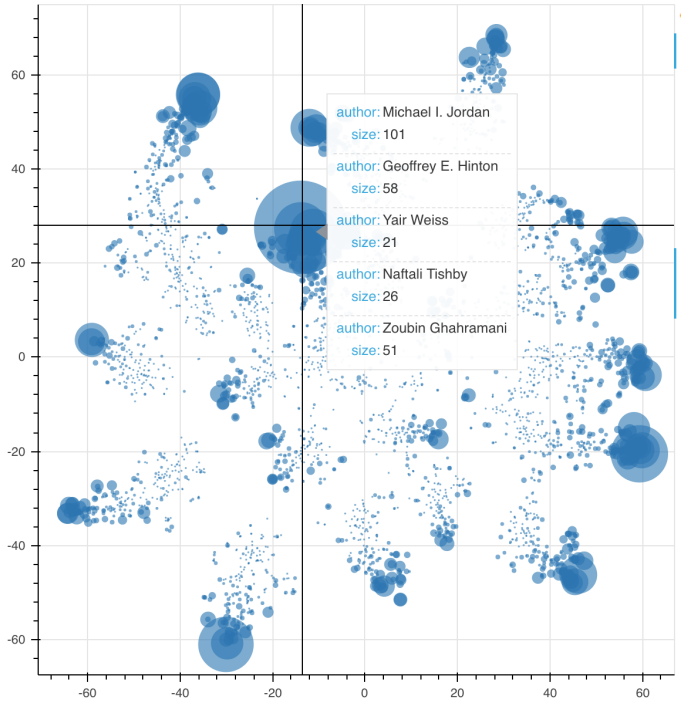

Fig. 7. Overview of authors' topics preference.

At last, we attempted to measure numerically the topic preference similarity between two authors. We use the Hellinger distance to measuring the distance (i.e. simi-

larity) between two authors's topic distribution (i.e their topic preference). Helligner distance measures the similarity between two probability distributions.The higher the Hellinger distance, the more dissmilar two distributions are. Its discrete version is defined as:

$$H(p,q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{K}(\sqrt{p_i} - \sqrt{q_i})^2} \quad (1)$$

where p and q are both topic distributions for two different authors. We define the similarity as follows:

$$S(p,q) = \frac{1}{1 + H(p,q)} \quad (2)$$

For instance, we would obtain the following table if we want to know who are the authors that are researching in the same areas as Terrence J. Sejnowski: The score column

	Author	Score	Size	topic
2845	Terrence J. Sejnowski	1.000000	48	[(11, 0.9999290468440801)]
2258	Peter Dayan	0.999995	47	[(11, 0.9999435699752137)]
534	Christof Koch	0.999985	35	[(11, 0.9998864424808276)]
2680	Si Wu	0.999912	9	[(11, 0.9996814870352174)]
66	Alan F. Murray	0.999907	11	[(11, 0.999664936787143)]
251	Anthony M. Zador	0.999897	8	[(11, 0.999638815817049)]
523	Christian K. Machens	0.999875	7	[(11, 0.9995754359247159)]
2469	Rodney J. Douglas	0.999872	8	[(11, 0.9995657793713213)]
509	Chris Diorio	0.999842	7	[(11, 0.9994811398151168)]
1620	Kevin A. Archie	0.999781	3	[(11, 0.999309748093670 1)]

indicates the similarity $S(p = \text{Terrence J. Sejnowski}, q)$, where q represents the other authors. The higher the score, the more similar they are. The topic column indicates the topic preference of this author. As we can see, they are all more or less interested to the topic 11. If Terrence was interested in different topics, the values of the topic columns would look like: $[(13, 0.6), (4, 0.3), (2, 0.1)]$.

VII. Additional Exploration

In the previous section, we have extracted the topics using the ATM model. As both LDA and ATM are probabilistic models, the key words for each topic generated from LDA might differ to the ones found by ATM. Indeed, we also tried to derive the topics using the LDA models, and have found similarities and differences compared to the ones we found using ATM. We gave each topic a named label for convenient description. We have listed the top-20 most frequent topics extracted using LDA in Table

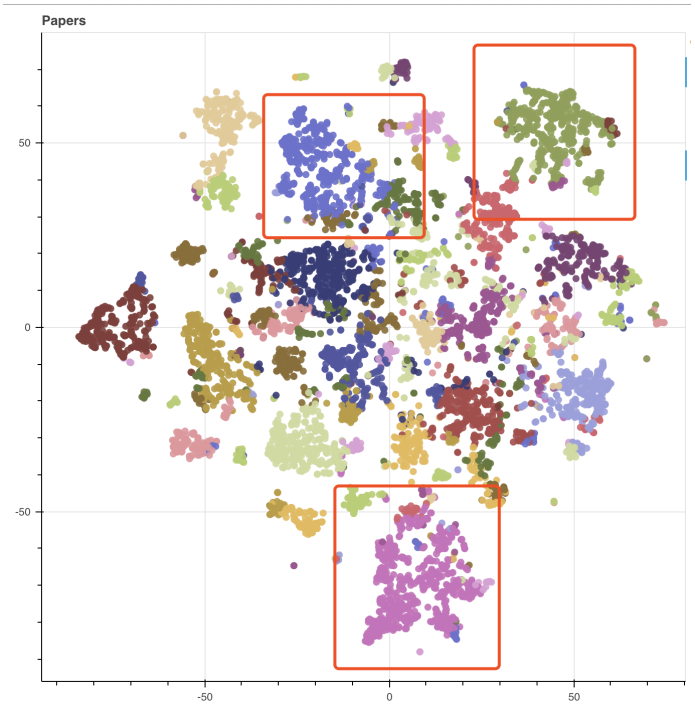

Fig. 8. Overview of the distribution of topics of papers.

II. After that, t-SNE is also applied for the exploitation of the distribution of topics of papers over years (Figure 8).

The 3 largest clusters of topics are Deep Learning, Gaussian Process and Reinforcement Learning. These seem to be the main research topics of the researchers participating to the NIPS conference.

VIII. Conclusion

In this project we have used a network-based approach to explore the 1987-2018 NIPS paper dataset. We have build two networks, the co-authorship network and the two version of topic networks using LDA and ATM models. We also gave a visualisation of our results using t-SNE. We had several findings about the NIPS papers. First, a lot of the authors who published several papers have co-authored with several other researchers, and as a result, we get a very large connected component. Few of them have a very large influence in their own field of research. The topics of NIPS papers are also derived. Using ATM, we tried to match the authors to the topics based on the papers they have written. From that, we attempted to re-discover the co-authorship through the topic preference of each user. At last, we have also tried to discover the similarity between researchers in terms of their research orientation.

REFERENCES

[1] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 05 2003.

[2] B Walsh. Markov chain monte carlo and gibbs sampling. *Lecture Notes for EEB 581, version 26, April*, 01 2004.

[3] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 07 2012.

[4] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008.

APPENDIX

Topic Label	High-frequency Terms									
Supervised Learning	label	classifier	margin	active	hypothesis	target	classifi	multi	boost	binary
Optimization	gradient	convex	objective	stochastic	dual	min	smooth	minimization	norm	converge
Gaussian Processes	posterior	bayesian	inference	likelihood	mixture	latent	variational	conditional	probabilistic	
Signal Processing	signal	filter	source	frequency	brain	channel	coefficient	correlation	speech	time_serie
Information Theory	estimator	density	finite	entropy	bias	continuous	asymptotic	theoretical	converge	uniform
Reinforcement Learning	policy	action	reward	agent	game	reinforcement_learn	player	environment	transition	strategy
Deep Learning	layer	deep	architecture	neural_network	convolutional	preprint	cnn	recurrent	gradient	rnn
Deep Learning	rule	neural$_network$	net	layer	hide	architecture	generalization	dynamic	activation	hidden_unit
Graph theory	graph	node	tree	edge	vertex	inference	partition	path	product	degree
Matrix and Tensor Factorization	sparse	column	rank	norm	sparsity	row	tensor	entry	recover	low_rank
Unsupervised Learning	cluster	distance	metric	similarity	manifold	embed	clustering	projection	nearest_neighbor	eigenvector
Computer vision	object	pixel	segmentation	recognition	face	detection	visual	scene	region	shape
Bandit algorithms	bind	regret	lemma	online	lower_bound	round	arm	upper_bound	bandit	strategy
Kernel methods	kernel	regression	svm	regularization	selection	operator	nonlinear	support_vector	polynomial	functional
Human Learning	human	response	trial	target	subject	stimulus	visual	behavior	cue	account
Collaborative filtering	user	rank	item	group	query	score	event	individual	link	preference
Information Retrieval	word	topic	document	language	text	sentence	lda	length	character	semantic
Naviguation and Planning	motion	position	location	region	trajectory	direction	field	move	surface	center
Neuroscience	neuron	cell	splike	activity	response	stimulus	circult	synaptic	dynamic	population
Distributed Computing	memory	search	bit	distribute	code	parallel	block	communication	scheme	store

TABLE II
TOP 10 HIGH-FREQUENCY TERMS FOR EACH TOPIC IN THE 20-TOPIC MODEL