Research Group Social Computing
Department of Informatics
Technical University of Munich

Social Topic Distributions

Mürüvvet Hasanbaşoğlu, Hakan Akyürek
Department of Data Engineering and Analytics & Informatics, Technical University of Munich (TUM)
✉ muruvvet.hasanbasoglu@tum.de, hakan.akyuerek@tum.de
30 July 2020

Abstract — Over the last decade the amount of textual data to work on increased rapidly due to the introduction and heavy usage of social media. Users in each social media network show different behaviours when writing comments to articles or posts. Ideally, users write comments if they are closely related to the topic of the respective article. In this project, we worked on the organic food dataset, where there are written articles or posts collected from multiple social media networks about organic food. We have provided our findings about how the users and articles are distributed among the sub-topics of organic food and other topics using Gaussian Mixture Models(GMM) learning technique with word and sentence vector representations. Additionally, based on a subset of articles from major social media networks, we checked whether the relatedness of the articles with the answering users compared to randomly selected users differ. Overall, results show that sentence vector representations provided better topic distributions of users and the answering users show closer topic distribution to the respective articles than the random users.

1 Introduction

The amount of available textual data is rapidly increasing as people start using the internet to share their thoughts and interests more and more. Even though the authors often leave tags on their written articles to define the related topics of their article, there might be still missed hidden topics, or users of the same social media resources do not even specify at which topic they are interested in, and they often leave comments either that support or oppose the thoughts of the author.

Topic modeling is a popular field of research in Natural Language Processing(NLP) and it is challenging to define and manually maintain the topics of the articles and user comments especially in social media resources [1]. There would initially be a need to extract the topics in an unsupervised manner, so that enhanced supervised techniques might be used to automate topic extraction from textual data. Therefore, topic modeling is referred as an unsupervised learning technique yet to perform document clustering. However, an article or a user comment often cannot be limited to one cluster where a cluster stands for a topic, but many different clusters or topics.

In this project, we would like to extract topics and do topic modeling on social media, forums and news websites articles and the user comments using vector representations of words and sentences and GMMs [2]. Working with GMMs means that we are dealing with probability distributions of clusters which tells us with how much probability does a document belong to which cluster. This way, we can extract the sub-topics discussed in the articles and also the sub-topics that the users of the websites are interested in and discuss about. This way, we can observe how the users and article authors in our whole dataset behave, how many talk about the main topic, organic food, how many talk about off-topic things and so on. We also would like to compare the probability distributions of articles and users who specifically comments on an article and additionally compare the probability distributions of articles and a random user who potentially is the user of the website and commented on different articles but not specifically to the mentioned article. Our assumption that topic distributions between the article and the answering users are closer compared to the topic distributions between the article and the random users. The intuition behind this is basically that the users often comment on the articles or posts only if they have an opinion about or interested in the topic of the article and we would like to analyze this with different sets of articles and users where users are both answering users and randomly selected users.

In this report, we would like to outline the approach we have taken to build a topic modeling scheme. In section 2, we will start with explaining the datasets used for training and testing, statistics of the datasets and applied pre-processing steps. In section 3, we dive into the implementation details explaining the vector representation extraction for words and sentences, how we have computed

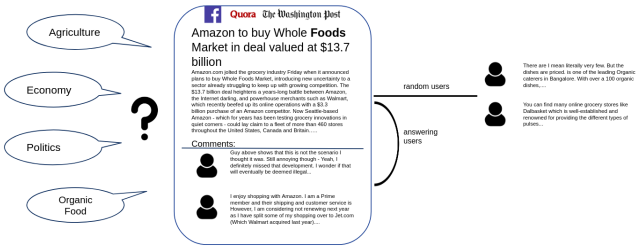

Figure 1 Illustration of our motivation

the optimal number of topic clusters and the topic labels as well as how we have extracted document level probability distributions for both words and sentences. In section 4, we list our experimental results about article and user clustering and showing the comparisons of topic distributions of answering users versus random users. Lastly, section 5 will provide an overall conclusion of our work.

2 Data

The dataset that is used in this project is called Raw Organic Dataset and is initially generated by the 1st generation of SocialROM students. We worked on an improved version called Curated Organic Dataset where word spellings are corrected, contractions are separated and language curation is applied by detecting and removing Hindi language [3]. The language of the dataset is English and it has multiple data resources categorized in two as biased and unbiased. Biased datasets contain users with biased opinion towards organic food and unbiased datasets contain users with general opinions. The categorization of the datasets is illustrated in Table 1.

The datasets are structured in JSON format where a JSON object includes an article text and the user comments with user ids and names that helps us identify a user within websites. Potentially, a user of a website writes many comments in multiple articles and a user can therefore be defined by concatenating his/her comments. We used the entire dataset to train our models and defined a subset to report our results mentioned in section 4. The subset is constructed by selecting 150 articles from Facebook, Quora and Washington Post respectively due to the high number of articles and users. The reasoning behind 150 articles is that our dataset also includes a relevance flag for each article indicating that the article is about organic food and Quora contains at most around 75 irrelevant articles. Therefore, we constructed the articles for results in a way that 75 of them are about organic food or food in general and 75 of them are about random topics. We also have different user datasets. First one is the answering users data which obviously indicates the users who replied to these 150 articles. Secondly, we have different random users dataset to take into account the category specifications of our datasets. In general random users, users are randomly selected from all around the dataset. In biased random users, the users are selected randomly among the biased datasets paying attention to that they are not answering user. In unbiased random users, the users are randomly selected among the unbiased datasets. In forum random users, the users are selected among forums datasets and in news sites random users, the users are selected among the news sites datasets.

Curated Organic Dataset			# of Articles	# of Users
Biased		Facebook	5,013	4,705
		Food Babe	15	15
		Food Revolution	78	60
		Organic Authority	66	0
		Organic Consumers	64	0
Unbiased	Forums	Cafe Mom	86	85
		Disqus	36	36
		Quora	567	523
		Reddit	81	78
		US Message Board	0	0
	News sites	Chicago Tribune	2,283	78
		Huffington Post	880	0
		LA Times	1,522	77
		NY Post	106	0
		NY Times	438	137
		USA Today	95	22
		Washington Post	1,563	943

Table 1 Dataset statistics with number of articles and users per data source

Due to the reason that the Curated Organic Dataset is composed of social media network and forum articles and the probability that the users do not pay attention to the proper usage of the language when writing comments, we performed additional pre-processing techniques as listed below:

1. Tokenization is performed with a regular expression where all the characters except all small and capital Latin alphabet, numbers, pe-

riod, question mark and exclamation mark are removed.
2. All characters are lower-cased.
3. Stop words are removed.
4. The words that passes at most 2 times in the entire dataset are removed.

The total number of tokens and the vocabulary size after the pre-processing steps is shown in Table 2.

Tokens	Vocabulary
∼19M	88,119

Table 2 Dataset vocabulary size

3 Implementation

3.1 Word embeddings

For vector representations of words, Common Crawl 1.9M pre-trained GloVe embeddings [4] were used due to the least number of out-of-vocabulary words. In total, 7013 words did not have embeddings in the pre-trained model and therefore, a fine-tuning was performed following the steps listed below [5] using the Mittens library [6]:

1. Load Common Crawl 1.9M pre-trained GloVe embeddings.
2. Compute the co-occurrence matrix for out-of-vocabulary based on the corpus of word list.
3. Train the embeddings for out-of-vocabulary using the co-occurrence matrix and pre-trained embeddings.

3.1.1 GMM training

The crucial point in topic modeling is to find the optimal number of topic clusters that is potentially discussed in the documents. Especially in the scope of GMMs, it is possible to compute different Gaussian topic distributions when the training is performed in different times using the same data and the same model parameters, because the expectation maximization algorithm that is used in GMM training potentially might lead the cluster means to result in different local minimas. Therefore, we used a method to observe how the GMM training reacts to training our document base and observe how stable the GMM training is with increasing number of clusters. The method essentially checks for the distance between two GMM models that are trained at the same time with the same configurations on the same dataset [7]. The training is also more automated to help us observe the training stability. After the range of the number of topic clusters was defined, 2 GMM models were trained 10 times per number of cluster and the distances between the GMM distributions was computed by the Jensen-Shannon divergence measure. Afterwards, the most closer 5 distances were selected per number of cluster. Figure 2 summarizes the means of the closest 5 distances per number of cluster and the vertical lines indicate the standard deviation of the 5 distances.

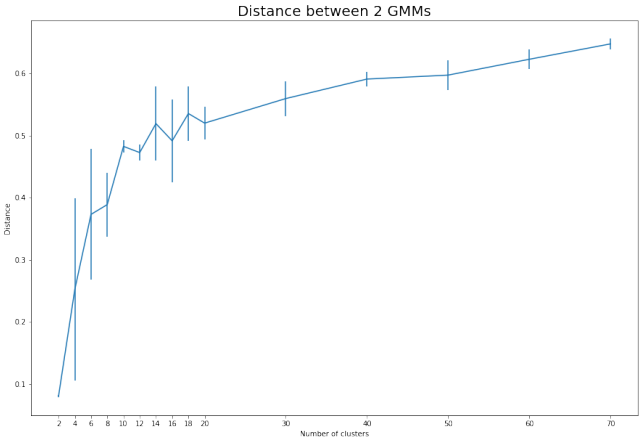

Figure 2 Distance between 2 GMMs trained with word embeddings

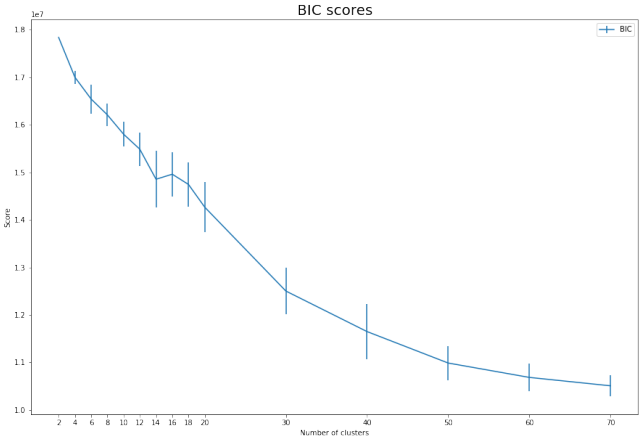

Figure 3 BIC Scores of GMMs trained with word embeddings

Next to this, the Bayesian Information Criterion(BIC) score was also computed at a training time per number of clusters using the same automated method. Figure 3 explains the mean of 5 best BIC

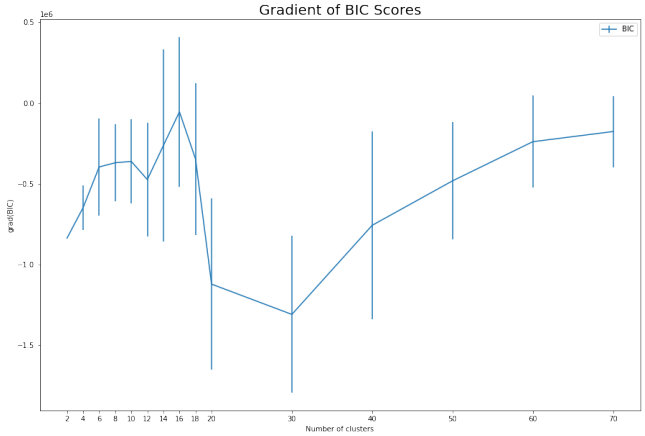

Figure 4 Gradient of the BIC Scores of GMMs trained with word embeddings

scores and the vertical lines refer to the standard deviation of the 5 best BIC scores per number of clusters. In Figure 3, it can be observed that the mean of the BIC scores continuously decreases as the number of clusters increases. To be able to decide a number, it is more intuitive to observe the gradient of the BIC scores. The gradient of the BIC scores is illustrated in Figure 4.

Figure 2 shows that the distance between 2 GMMs follows a logarithmic function. As a result, it can be said that after a number of clusters, the distance between two GMMs does not change enormously and it starts increasing steadily. Also, the variations between distances of different trainings is more stable starting from 20 clusters and quite low compared to lower number clusters. 20 seems to be a good decision point in Figure 2. However, Figure 3 shows that there is a wave around 14 and starting from 18, the score steadily decreases. Looking at the gradients of the BIC scores in Figure 4, after 30 clusters there is a stable increase, the gradient is getting closer to 0, which means the decrease in BIC score loses importance after 30 clusters. It can also be observed in Figure 4 that around 16 clusters, there is a near 0 gradient. Based on these interpretations, we decided on selecting 15 and 30 as number of clusters and the GMM trainings for both configurations was performed using the entire word embeddings.

A GMM trained with word embeddings simply gets a word's embeddings as an input and outputs the probability distribution of the word. In order to obtain the probability distribution of a document from the probability distributions of the words in that document, we apply the following procedure: Firstly, the CF-IDF technique[8] is applied to the probability distributions of the words, so that if a concept occurred in many documents, that concept is weighted low and if a concept has occurred in rare documents, that concept is weighted high. A concept in our case refers to a cluster. Our goal using this weighting is to push the probability of those clusters which includes many general or garbage words low. The CF-IDF is calculated using the following formula where $\text{CF}(c_i, d_j)$ is the column-wise sum of the probability distributions of each word of within a document.

$$\begin{aligned} &\text{CF-IDF}(c_i, d_j, D) \\ &= \text{CF}(c_i, d_j) \times log \frac{|D|}{|d \in D; c_i \in d|} \end{aligned} \tag{1}$$

where $|D|$ is the number of documents in the corpus, $|d \in D; c_i \in d|$ is the number of documents in the corpus with concept c_i, and (c_i, d_j, D) is $concept_i, document_j, corpus$ respectively. Then, in order to find the j'th document's probability distribution, we use the following formula:

$$P(c_i, d_j, D) = \frac{\text{CF-IDF}(c_i, d_j, D)}{\sum_{m=1}^{n} \text{CF-IDF}(c_m, d_j, D)} \tag{2}$$

where n is the number of concepts, k is the number of words in the document, w_l denotes the CF-IDF applied probability distribution of l^{th} word in the document, w_{li} is the CF-IDF applied probability for l^{th} word for concept i.

3.1.2 Topic Labeling

Topic labeling is another crucial point in topic modeling studies because it is mostly used as an evaluation criteria for the generated clusters. Once a set of representative list of words are extracted from the clusters, we can evaluate how coherent the clusters are. Thereafter, to manually assign a topic label for each cluster is commonly used technique to define the topics for clusters [9].

When working with word embeddings, we used cosine similarity to find the closest words to each Gaussian mean. Gaussian stands for a cluster or a topic in the context of GMM. More specifically, we find the top 20 closest words for each Gaussian that are labeled as the respective Gaussian. The formula is defined as follows:

$$D = \frac{W \cdot M}{\|W\| \times \|M\|} \tag{3}$$

where W is the normalized word embedding, M is the mean of the respective Gaussian, and D is the similarity between them.

Generally, we observed a lot of garbage or general clusters with word embeddings and it was more difficult to observe a specific coherence between the words compared to the sentence embeddings which is mentioned in later sections. The GMMs trained with word embeddings captured some out-of-topic aspects as well. Table 3 gives a brief summary of the cluster labels:

Label	Top Terms
Prompter	totus prompter teleprompter
Food Ingredients	spinach salad onions cheese
Biology	acids metabolites enzymes proteins microorganisms
Medical	disease symptoms chronic disorders infections
Garbage	patelin ijpba michaeleisen kopley
Sexual Content	fuck slut pussy
Laughter	hahah hahahaha lmao lolol
Biomes & City	ashland riverside cumberland burlington
Politics & Religion	christians government political war muslim

Table 3 Highest scored words for some of the clusters

3.2 Sentence embeddings

For vector representations of sentences, sentence retrieval and sentence pre-processing was applied initially. The sentences that included only punctuation and one word were removed as well as duplicate sentences and sentences that contain less than 15 characters. There exist in total around 1.6M sentences in Curated Organic Dataset. Initially, CoLab's working environment was being used until we came across with many run-time crashes due to insufficient RAM capacity. Often times, we tried to configure the implementation in a way that sentences are trained in batches. However, this approach also did not work due to the long lasting computation time estimations and CoLab's at most 12 hours connection limitation. Due to these reasons, sentence embeddings generation, experiments with sentences and the results of the models using sentence embeddings are run in Workstation Server social4. After the necessary hardware capabilities were provided, sentence embeddings were generated using Universal Sentence Encoder Version 4 with dimension 512. [10]

3.2.1 GMM training

To find the optimal number of clusters with sentence embeddings used, we repeated the same experiments done in section 3.1.1. However, running the automated method using the entire set of sentences was expensive due to the computation time. Therefore, a subset was defined by randomly selecting 100k sentences from the entire set of sentences and the range of number of clusters was narrowed. Figure 5 summarizes the means of the closest 5 distances of 2 GMMs per number of cluster and the vertical lines indicate the standard deviation of the 5 GMM distances. Figure 6 explains the mean of 5 best BIC scores and the vertical lines refer to the standard deviation of the 5 best BIC scores per number of clusters. In Figure 6, it can be observed that the mean of the BIC scores continuously decreases as the number of clusters increases just like the case with the word embeddings. To be able to decide a number, we observe the gradient of the decrements and the gradient of the BIC scores is illustrated in Figure 7.

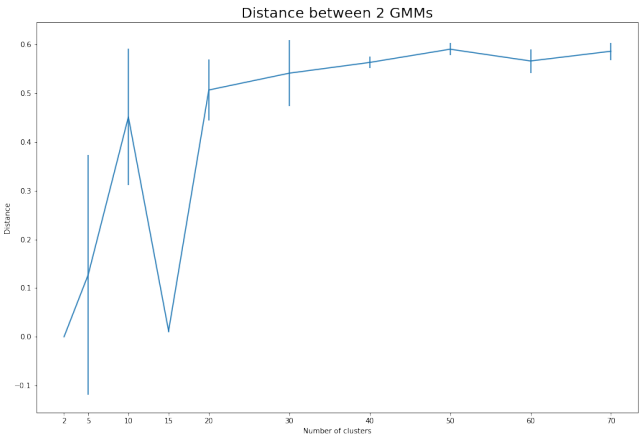

Figure 5 Distance between 2 GMMs trained with sentence embeddings

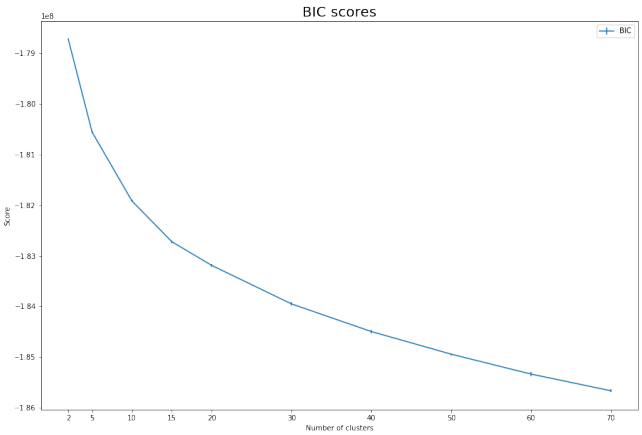

Figure 6 BIC Scores of GMMs trained with sentence embeddings

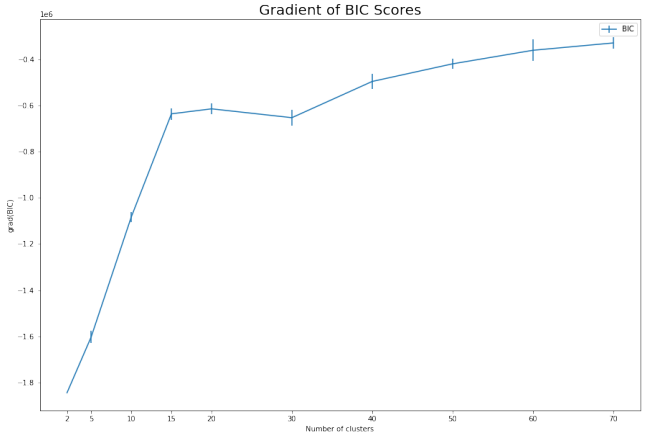

Figure 7 Gradient of the BIC Scores of GMMs trained with sentence embeddings

Figure 5 shows that the distances between 2 GMMs are more or less the same starting from the 20 number of clusters. The distances show a big variance between trainings where the number of cluster is smaller then 20, which means the GMM trainings with number of clusters smaller than 20 are extremely unsteady. Figure 7 shows that starting from 15 the gradient of the scores increases much slower than before 15. Both of the figures show that the number of clusters should be larger than 15, because the scores and the distances have big variances among different timed trainings and sudden decrease on the BIC scores before 15. In Figure 7, it can also be observed that starting from 30, there is a constant increase in the gradient of the BIC scores. Due to the combination of all the reasoning same as the word embeddings, we wanted to observe the results of our training using 15 and 30 number of clusters and therefore, 2 GMM trainings were performed with 15 and 30 clusters using the entire set of sentence embeddings that we derived from our dataset.

As in the case of word embeddings mentioned in 3.1, the GMMs that are trained with sentence embeddings takes a sentence's embeddings as an input and provides a probability distribution for the sentence which tells with how much probability does the sentence belongs to which topic cluster denoted as $W^{(j)} \in R^{k \times n}$ where k is the number number of sentences in a document and n is the number of topic clusters. In order to provide a text level probability distribution for the j^{th} document, we use the following formula:

$$P(c_i, d_j) = \frac{\sum_{l=1}^{k} w_{li}^{(j)}}{\sum_{m=1}^{n} \sum_{l=1}^{k} w_{lm}^{(j)}} \tag{4}$$

Simply, the probability distributions of sentences within a document is summed up per cluster to get the probability distribution of the text and then the distribution is normalized so that the probabilities sum up to 1.

3.2.2 Topic labeling

In order to label the topics, the approach of finding the best representative n words of each cluster was applied as in section 3.1.2. However, when working with sentences, clarity scoring function was observed to produce pleasant results to extract the representative list of words of the clusters named aspect extraction [11].

$$clarity\_score_a(w) = t_a(w) \log_2 \frac{t_a(w)}{t(w)} \tag{5}$$

In order to achieve this, we firstly predicted for which cluster does a sentence have the maximum probability in the entire set of sentences. Once we obtained the clustered sentences, the function 5 is applied where $t_a(w)$ is $l1$ normalized TF-IDF values of sentences within a cluster a and $t(w)$ is the $l1$ normalized TF-IDF values of sentences among the entire set. Finally, 20 words with highest clarity score was selected to define the top 20 representative list of words per cluster. Table 4 summarizes some of the top n words and the manually inferred topic labels. From the top terms and the labels, we can interpret that the clusters seem to be more coherent and within the topic with trained GMM using sentence embeddings.

Label	Top Terms
Agriculture	farm soil grow cattle chickens
GMO	gmo monsanto seeds corn labeling engineered
Food Ingredients	milk cheese butter chicken oil
Science & Faith	god science evolution existence religion
USDA Certified Organic Food	organic certified product usda standards
Gardening	plant bees garden soil flowers fertilizer
Economy & Politics	tax money federal capitalism income wage
Global Warming	climate warming global carbon co2 planet
Grocery Stores & Shopping	store products walmart buying food amazon

Table 4 Highest scored words for some of the clusters

4 Experimental Results

We implemented and trained GMMs as we described in section 3 using Scikit Learn library [12]. As a result, we trained 4 GMMs training using word embeddings and sentence embeddings in 15 and 30 number of clusters configurations. Thereafter, we report various experiments using these models. In our experiments, we focused on 3 main questions:

- How are the articles distributed in each of the GMMs?
- How are the users distributed in each of the GMMs?
- Do the answering users have closer distributions to the articles compared to the random users?

For the clustering experiments, the extracted labels from the clusters with methodology mentioned in Topic Labeling sections can be seen in the visualizations of each clustering experiment.

4.1 Article Clustering

The entire set of articles are initially predicted using GMMs to be in which cluster with how much probability. Thereafter, the maximum probability was selected to be the topic cluster of the specific article.

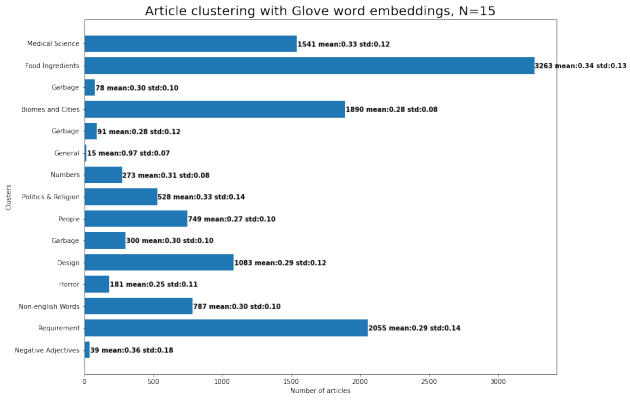

Figure 8 Article distributions in GMM with 15 clusters and word embeddings

Figure 8 and Figure 9 show the distribution of the articles in our dataset when we used word embeddings to train the GMM. The y-axis shows the topics in the GMM and the x-axis show the amount of articles that belong to the clusters or topics. Right to the bars, the means and standard deviations of the

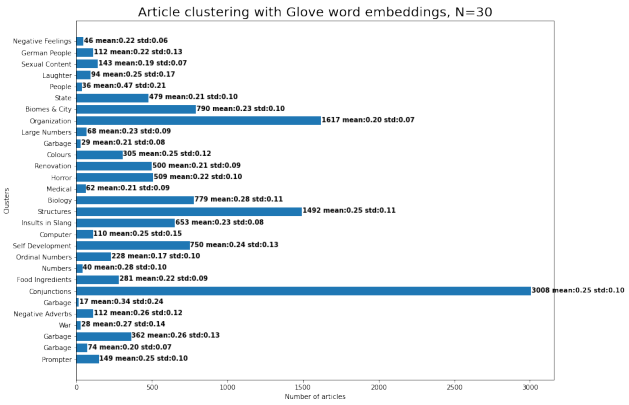

Figure 9 Article distributions in GMM with 30 clusters and word embeddings

probabilities of articles that are assigned to the respective cluster are indicated.

In 15 clusters model using word embeddings, most of the articles appear to be belonging to on-topic clusters and the distribution of the articles proved to be balanced. For instance 3263 articles belong to the cluster about food ingredients and least amount of articles belong to topics about general words or garbage clusters. Similarly, we observed a balanced distribution in 30 clusters model using word embeddings. However, the GMM with 30 clusters could not achieve the meaningful article distribution that GMM with 15 clusters achieved, the articles belong to more general and out-of-topic clusters. For example, the cluster with most articles is 'Conjunctions' cluster with 3008 articles and the following peaks are about organizations and structures. These results show that 30 clusters GMM was able to have more discrete and specialized clusters, but in that case the articles tend to belong to garbage and general clusters.

Additionally, it is worth noting that with CF-IDF, we were able to achieve more balanced distributions and without CF-IDF the distributions were extremely imbalanced. However, Figure 9 shows that even with CF-IDF, the distribution of articles is imbalanced and the articles are clustered in general or garbage topics.

Figure 10 and Figure 11 show the distribution of the articles using 15 and 30 clustered GMMs trained using sentence embeddings. The results show that with 30 clusters, we can observe a more balanced distribution of articles, while with 15 clusters it is more imbalanced. The peak distributions are the

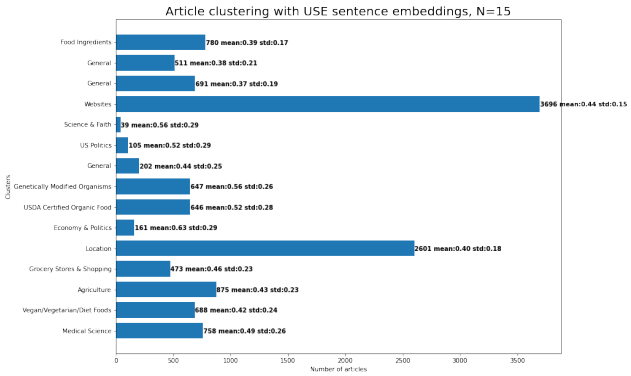

Figure 10 Article distributions in GMM with 15 clusters and sentence embeddings

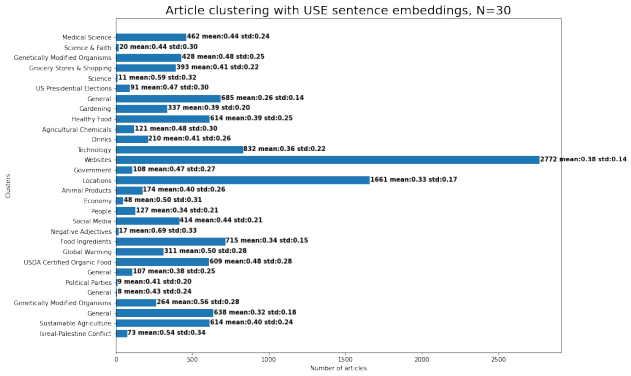

Figure 11 Article distributions in GMM with 30 clusters and sentence embeddings

same in both configurations about locations and web sites.

To sum up, we can observe from the article clustering results that our models have high confidence when assigning articles to the respective clusters. The articles are often written with a more careful language compared to the user comments and we can clearly observe there are few articles in all the models that are assigned to the garbage clusters or general clusters which does not indicate a topic but constructed from general/meaningless words.

4.2 User Clustering

Similarly, the entire set of users are initially predicted using GMMs to be in which cluster with how much probability. Thereafter, the maximum probability was selected to be the topic cluster of the specific user where a user is constructed by concatenating that users comments.

Figure 12 and Figure 13 show the distribution of the users in our dataset when we used word em-

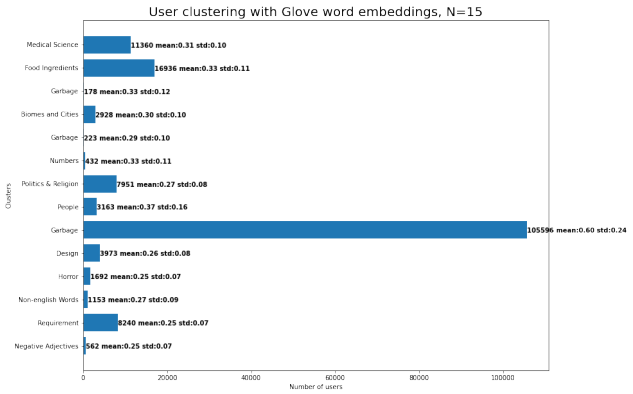

Figure 12 User distributions in GMM with 15 clusters and word embeddings

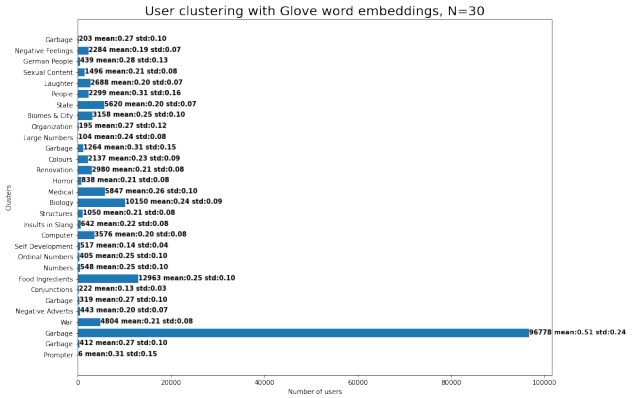

Figure 13 User distributions in GMM with 30 clusters and word embeddings

beddings to train the GMMs in 15 and 30 number of clusters respectively. Both with 15 and 30 clusters, the GMMs were not able to provide a balanced user distributions. Most of the users tend to cluster into one of the topics which is a garbage cluster for both of the distributions. Also, some number of users are clustered in topics about food ingredients, medical science or biology. After applying CF-IDF into word probability computation, we observed more balanced user distributions among clusters.

The GMMs trained using the sentence embeddings provided much more satisfying results. Figure 14 and Figure 15 show that with both 15 and 30 clusters, the user distributions are much more balanced and a huge amount of users cluster to on-topic clusters. However, in both distributions the peak clusters are general or garbage clusters with relatively high means and standard deviations, whilst the amount of users with non-garbage opinions are majority.

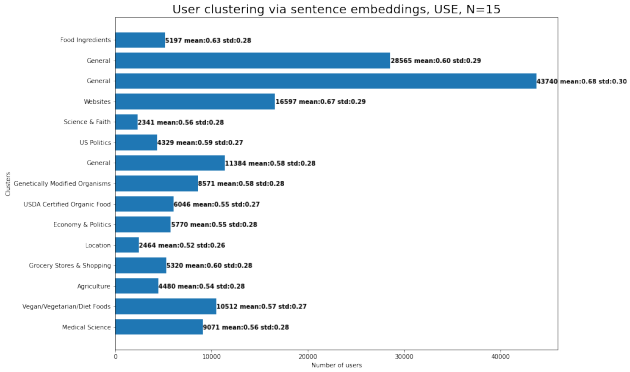

Figure 14 User distributions in GMM with 15 clusters and sentence embeddings

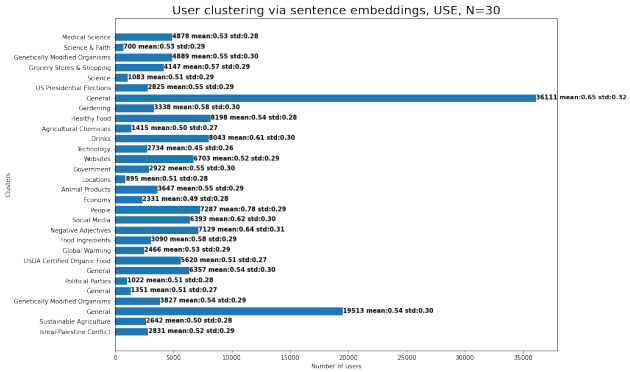

Figure 15 User distributions in GMM with 30 clusters and sentence embeddings

As a result, for clustering our experiments suggest that when we train our GMMs with sentence embeddings, we get more promising results. The distributions are more balanced in general for articles and users, meaning that the GMMs were able to capture better topic distributions when trained with sentence embeddings as also the cluster labels suggest. Additionally, overall means of assigned users and articles in GMMs trained with sentence embeddings tend to be higher than GMMs trained with word embeddings.

4.3 Answering users versus Random users

Lastly, we answer the question "Do the answering users have closer distributions to the articles compared to the random users?".

We calculate the distance between an user and article with Jensen-Shannon Divergence(JSD) with the following formula:

$$JSD(A\|U) = \frac{1}{2}D(A\|M) + \frac{1}{2}D(U\|M) \quad (6)$$

where A is the probability distribution of the article, U is the probability distribution of the user, D is the Kullback–Leibler divergence, and M is $\frac{1}{2}(A+U)$ and the square root of JSD gives us the distance between the two probability distributions.

To measure whether answering users are closer, we calculate the JS Distance between the articles where the articles are are selected from the datasets as mentioned in section 2 and their respective answering users. Later, we sample n random users from the random user sets, where n is the number of answering users for the respective article, and calculate the JS Distance between these users and the articles. Then, we calculate the mean and standard deviation of the distances of answering and random users. Equation 7 demonstrates the mean and standard deviation calculation for one article:

$$\begin{aligned} \mu_{au} &= \frac{\sum_{i=1}^{n} JSD(A\|AU_i)}{n} \\ \sigma_{au} &= \sqrt{\frac{\sum_{i=1}^{n}(AU_i - \mu_{au})^2}{n}} \\ \mu_{ru} &= \frac{\sum_{i=1}^{n} JSD(A\|RU_i)}{n} \\ \sigma_{ru} &= \sqrt{\frac{\sum_{i=1}^{n}(RU_i - \mu_{ru})^2}{n}} \end{aligned} \quad (7)$$

where AU_i is the i^{th} user of the answering users, RU_i is the i^{th} user of the random users. μ_{au} and σ_{au} denote the mean of answering user distances and standard deviation of answering user distances respectively, whereas μ_{ru} and σ_{ru} denote mean of random user distances and standard deviation of random user distances respectively. Moreover, we repeat this calculation for 150 articles from mentioned datasets and calculate the average of means and standard deviations over all the articles for one dataset. We do this operation for each model and user set combinations. Equation 8 defines this operation for one dataset:

$$\begin{aligned} \mu_{\mu_D} &= \frac{\sum_{i=1}^{N} \mu_u}{N} \\ \mu_{\sigma_D} &= \frac{\sum_{i=1}^{N} \sigma_u}{N} \end{aligned} \quad (8)$$

where N is the number of articles, D is the dataset, μ_u and σ_u are mean and standard deviation

of either random users or answering users, and μ_{μ_D} and μ_{σ_D} are average of distance means and average of distance standard deviations respectively.

Figure 16, Figure 17, Figure 18 show us the results for mentioned datasets. The y-axes shows the values for average means and standard deviations. The x-axes show the experiments different user sets. Each bar shows the average mean for a GMM; "30W" is 30 clusters with word embeddings, "15W" is 15 clusters with word embeddings, "30S" is 30 clusters with sentence embeddings, "15S" is 15 clusters with sentence embeddings. The black lines show the average standard deviations .

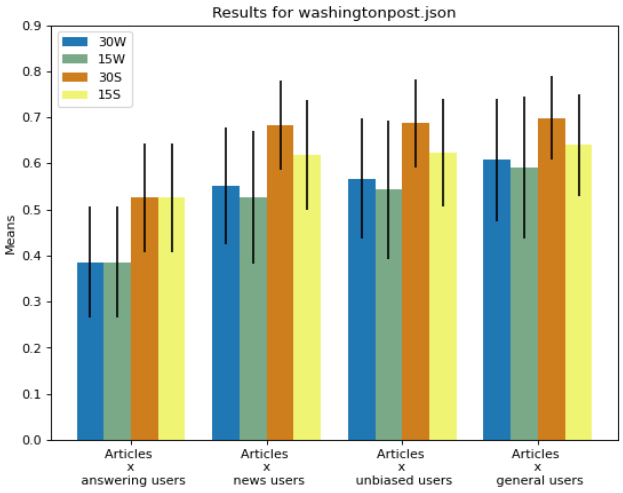

Figure 16 Results of Washington Post dataset

In Figure 16, it can be seen that the mean for answering users is smaller than the mean for each random user set. One point to note is that as the random user set gets more specific, e.g. news is more specific than general, the gap between the random user means and answering user means becomes smaller and the average standard deviations get bigger. This is expected since the users in news datasets are expected to have more similar behaviours to the users in Washington Post.

Figure 16 also shows that when we use 15 clusters with both sentence and word embeddings the gaps between means are smaller. This isn't actually a better result since the gap between means should be large, because we want to have more confidence when deciding whether a user is answering or random user.

Figure 17 shows that the GMMs achieve similar results with the articles in Quora dataset. However, the gaps are smaller than the gaps in Washington Post results. These results show us that the user

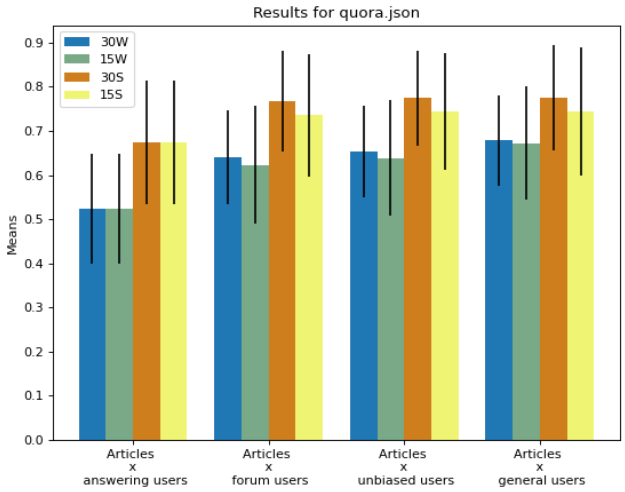

Figure 17 Results of Quora dataset

base of Washington Post actually tend to stick to the topic more than the users in Quora. Figure 17 also shows that the average means for both answering and random users are bigger in Quora dataset, which suggests that the articles of Quora are different than the articles in Washington Post and the user base of Quora actually do not stick to the topic as much as the Washington Post users.

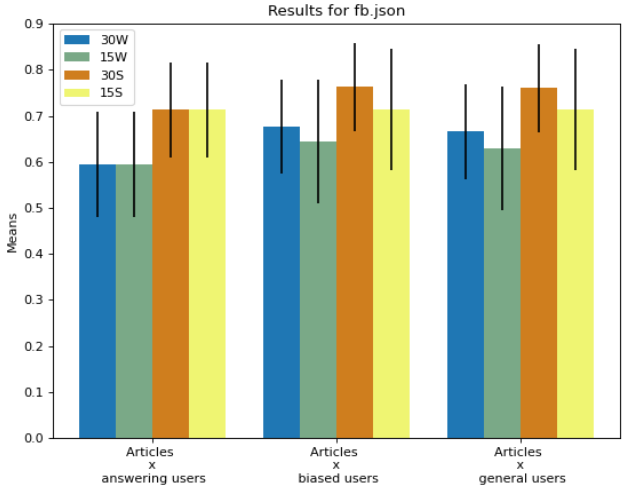

Figure 18 Results of Facebook dataset

Figure 18 shows that the GMMs cannot achieve the expected results with Facebook articles and users. The gaps are even more smaller and when 15 clusters are used, we observe really close average means with random users with respect to answering users. The results show that when sentence embeddings are used with 15 clusters the average means of random users are actually smaller

than average means of answering users. Also, with word embeddings and 15 clusters, Facebook does not follow the "more specific dataset, smaller gap" pattern. When random users are sampled from general set the gap is smaller with respect to when users are samples from biased set.

Figure 18 also shows that the users of Facebook do not tend to stick to the topic and the articles much more different than other datasets' articles. The average mean of answering users is also higher than Quora as well and the gaps between means is rather small.

It should be noted that in our experiments we got significantly different average means and standard deviations from sentence and word embeddings. However, since they explain the users and articles in different ways, comparing these results would be inaccurate and wrong.

5 Conclusions

In this project, we present various GMMs trained with different bases and with different configurations to solve social topic distribution tasks. Our models can successfully create balanced user clusters and article clusters with sentence embeddings. The answering user distributions are generally closer to the article distributions than the random user distributions. The GMMs with 30 clusters tend to show better results in all aspects. We believe that with 30 cluster GMMs, we are able to capture more aspects and better results. However, as the datasets are from different social media resources, there tend to be different topics, but still we were able to observe many specific topics about organic food.

On the other hand, word embeddings also show promising results. The user and article clusters aren't balanced, but answering user distributions are generally closer to the article distributions than random user distributions. However, we believe working with sentence embeddings show more reliable results than word embeddings.

Lastly, the results suggest the importance of users in the dataset. In general, answering user distributions are generally closer to the article distributions, however, corrupt user behaviours and the noisy data in the internet and especially in the social media come along with their challenges.

Acknowledgement

We would like to thank Gerhard Hagerer, M.Sc for the assistance and guidance with the methodology and the technical hardware support. We also would like to show our appreciation to the Research Group of Social Computing for initiating the organisation of the NLP Lab Course.

References

[1] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the first workshop on social media analytics*, 2010, pp. 80–88.

[2] V. K. R. Sridhar, "Unsupervised topic modeling for short texts using distributed representations of words," in *Proceedings of the 1st workshop on vector space modeling for natural language processing*, 2015, pp. 192–200.

[3] "Curated organic dataset." [Online]. Available: https://gitlab.lrz.de/social-rom/organic-dataset/curated-source-dataset/-/tree/master/english

[4] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[5] S. Santhanam, "Fine tune glove embeddings using mittens," Apr 2020. [Online]. Available: https://towardsdatascience.com/fine-tune-glove-embeddings-using-mittens-89b5f3fe4c39

[6] N. Dingwall and C. Potts, "Mittens: an extension of GloVe for learning domain-specialized representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 212–217. [Online]. Available: https://www.aclweb.org/anthology/N18-2034

[7] V. Lavorini, "Gaussian mixture model clusterization: how to select the number of

components (clusters)," Nov 2018. [Online]. Available: https://towardsdatascience.com/gaussian-mixture-model-clusterization-how-to-select-the-number-of-components-clusters-553bef45f6e4

[8] H. K. Kim, H. Kim, and S. Cho, "Bag-of-concepts: Comprehending document representation through clustering words in distributed representation," *Neurocomputing*, vol. 266, 05 2017.

[9] S. Syed and M. Spruit, "Full-text or abstract? examining topic coherence scores using latent dirichlet allocation," in *2017 IEEE International conference on data science and advanced analytics (DSAA)*. IEEE, 2017, pp. 165–174.

[10] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018. [Online]. Available: http://arxiv.org/abs/1803.11175

[11] S. Angelidis and M. Lapata, "Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised," *arXiv preprint arXiv:1808.08858*, 2018.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.