Webscraping and Text Analysis: Final Report

Supervised Reading Project Summer 20223

Abraham Morales

Introduction

In this project, we aim to analyze profiles of nannies offering their services across Canada. In this first iteration of the project, we focused our attention on the province of Ontario. Profile information is publicly available on the site CareGuide: CanadianNanny.ca [1]. This information was scraped using R, specifically the packages rvest and stringr. The goal of the project was to extract patterns from the descriptions of users and the services they were offering, as well as their location, years of experience, hourly rate, etc.
In recent years, there has been a lot of development in the field of Natural Language Processing (NLP). The problem addressed in this report adds a degree of difficulty, given that our data is completely unlabeled, and our main interest is to extract patterns from the text. We believe this is more in line with real-world applications, where labeling datasets through human input is often a monumental task impossible for most research teams or companies.

Data

As mentioned before, the dataset was scraped from publicly available profile descriptions of nannies seeking job opportunities in Ontario. The scraping script goes over a set number of pages on the Ontario section of the website [1], which has over 1000 pages with 10 profiles each. We were mostly interested in the text sections of the profiles. These include the short blurb right under the profile picture, a "Reasons to Hire Me" section, and a longer description in the "About Me" section.

We also retrieved the users' name, location, reported years of experience, hourly rate, last time active on the site, number of reviews, star rating out of five, bullet points under the "I can work:" subsection (part-time, full-time, summer), children ages the user has experience with (infant, toddler, newborn), number of children they can look after, experience with children with medical conditions (diabetes, disability, epilepsy, severe allergies, etc.), transportation requirements (close to transit, has driver's license), qualifications (first aid, CPR, languages, etc.), and services they can provide (housekeeping, cooking, groceries, swim supervision, etc.). The link to the users' profile and the date the information

was retrieved were also stored in the dataset to check for changes in the profiles (check data comparison notebook).

The scraping script was run at multiple time points during the month of June, and we ended up with around 6000 profiles, some of which might be the same one but retrieved on different days. For this comparison, the profile's link serves as a unique identifier.

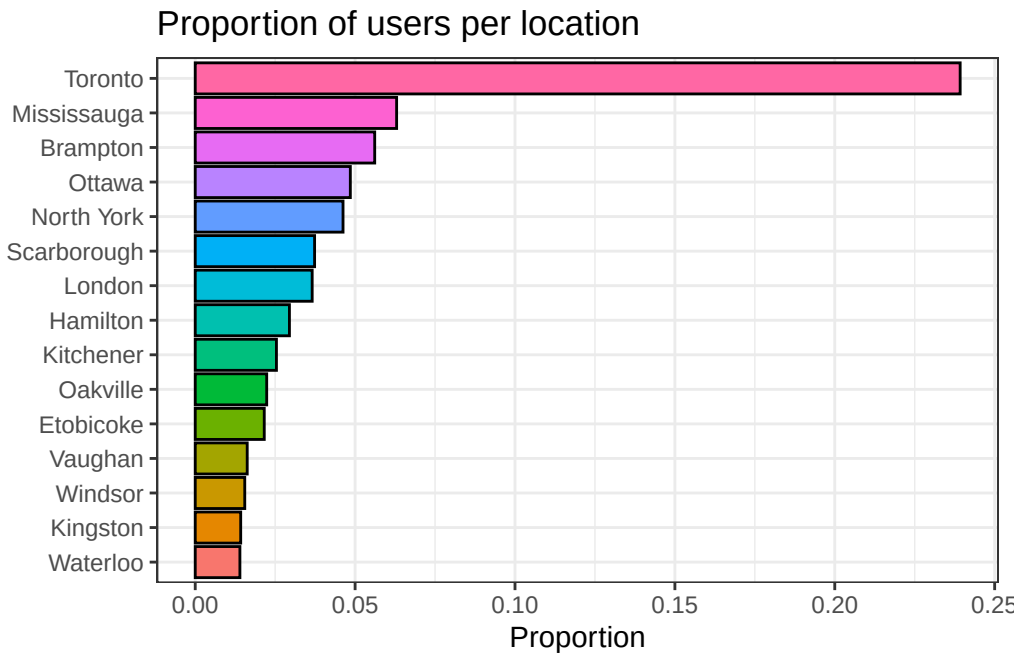


Figure 1: Percentage of users by location in Ontario.

Exploratory Data Analysis

For this work, we are only focusing on users located in Ontario. In Figure 1, we can see the proportion of users per location. The city of Toronto accounts for almost a quarter of all users, and that's not counting nannies who reported living in Scarborough, North York, or Etobicoke, which are also part of the Toronto municipality. Mississauga and Brampton follow in 2nd and 3rd place, respectively. Although these are independent cities, they're part of the Greater Toronto Area. Since most nannies are based in close proximity to one another, it might be difficult to detect notable differences in their profile descriptions purely based on location. If a spatial analysis is to take place, the whole country must be considered.

As mentioned before, each user describes on their profiles the services they are willing or qualified to provide. One thing worth exploring is how the requested hourly rate varies depending on the number of qualifications the nannies listed. Figure 2 shows the distribution of hourly rates by the number of

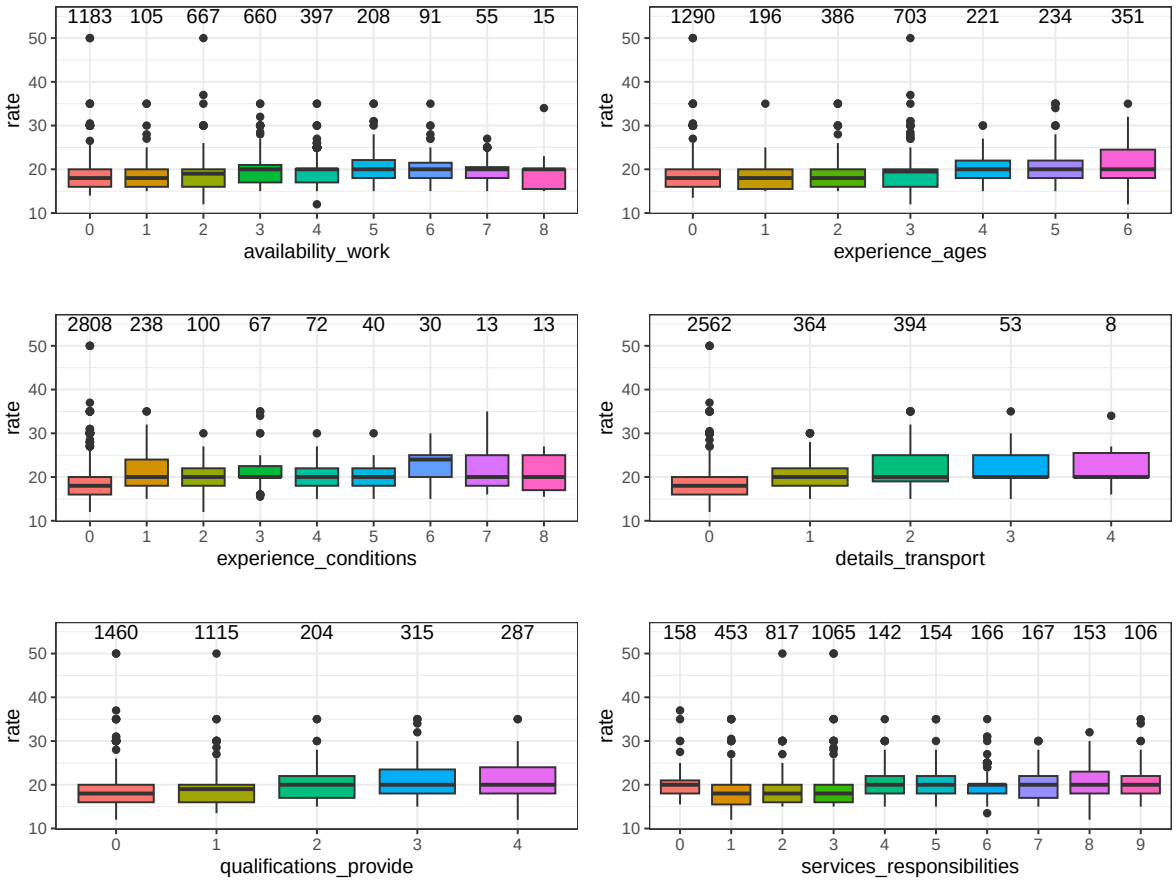


Figure 2: Distribution of requested hourly rates by number of items listed under different subsections.

items users wrote under those subsections. Above each boxplot, the number of users per category is shown. We can see that except for the subsection of "responsibilities" under "Services," those who didn't have items listed under that subsection had the lowest median rate. However, the difference between distributions isn't significant enough to draw any conclusions just yet.

Figure 3 shows the difference in rate distribution for users who reported being qualified to provide Cardiopulmonary resuscitation (CPR). We can see that those who can provide CPR tend to request a higher hourly rate on average. The same behavior was observed at multiple time points throughout the month of June.

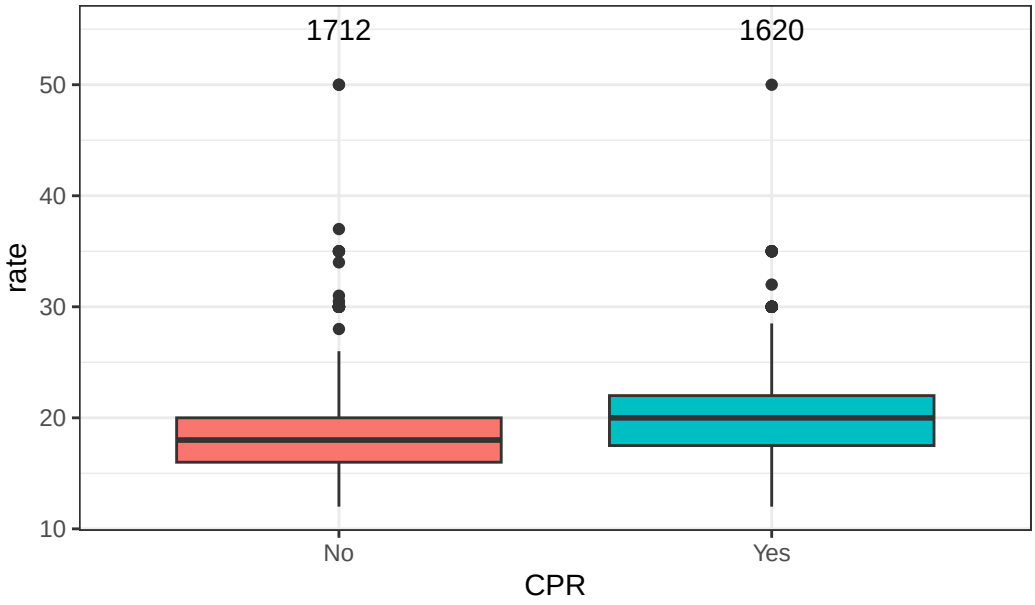


Figure 3: Distribution of hourly rates based on whether the user can provide CPR.

Exploratory Text Analysis

The profile descriptions of the users were highly similar to one another; all of them talked overwhelmingly positively about themselves, which isn't surprising since they are trying to establish trust with the parents. In a more in-depth text analysis, it was found that the users who had a negative sentiment score wrote a very short description and included phrases such as "don't hesitate to…". Most lexicons classify the word "hesitate" as negative. In Figure 4, we can see the most common words that appeared in the "About Me" section of the profiles. Unsurprisingly, the most common words were "children," "care," and "nanny."

We are also interested in how words are related to each other; that is, what are the most common words that come before a specific word. In Figure 5, we can see an arrangement of words in a network, or

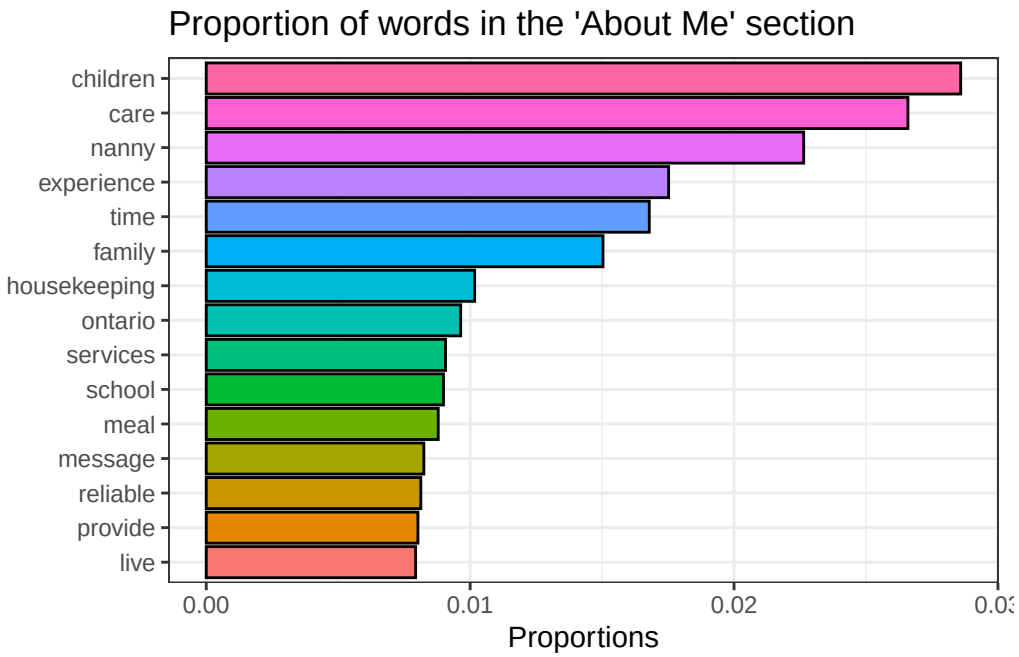


Figure 4: Proportion of appearance of words in the profile description (removing stop words).

"graph." This was done using the igraph and ggraph libraries [2-4]. Some of the most common bigram examples are "experienced/responsible/professional nanny," "pet care," "meal preparation," and "primary school." We can also see that the network detects the cities the nannies are based in when they mention it in their descriptions.

Text Classification Approaches

We used state-of-the-art deep learning and baseline models to create text representations (i.e., sentence and word vector embeddings). Then, we utilized Principal Component Analysis (PCA) to reduce the dimensions of our embeddings. We applied these reduced vectors to various clustering algorithms, including K-means, Gaussian Mixture Models, and DBSCAN, to generate topics using a similarity-based approach.

We compared this approach with statistical models for unsupervised text classification, such as Latent Dirichlet Allocation (LDA). Since our data is unlabeled, the performance analysis and comparison of different models were primarily subjective. Next, we will briefly explain the models that were used.

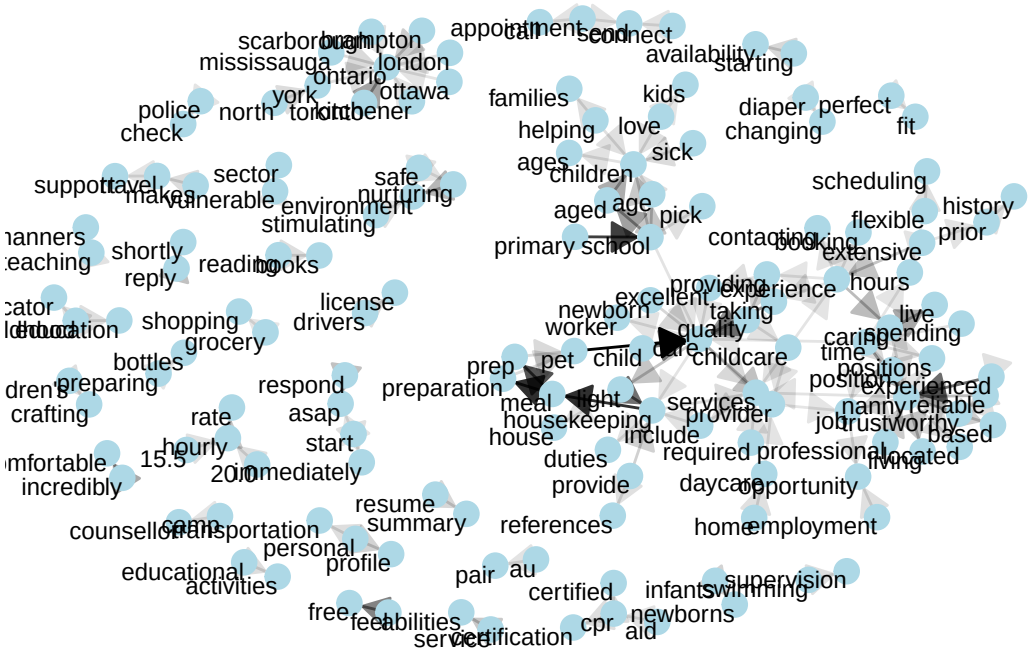


Figure 5: Graph representation of bigrams in the profile descriptions (removing stop words).

Latent Dirichlet Allocation

LDA is a generative probabilistic model where documents are assumed to be generated by a mixture of topics, and each topic is represented by a distribution over words. In this model, a document is a sequence of N words $\mathbf{w} = (w_1, \dots, w_N)$, and a corpus is a collection of M documents denoted by $D = \mathbf{w}_1, \dots, \mathbf{w}_M$. For each document in the corpus D, we set the following priors:

$$
\begin{aligned}
N &\sim \text{Poisson}(\xi) \\
\theta &\sim \text{Dirichlet}(\alpha) \\
z_n &\sim \text{Multinomial}(\theta), \quad \text{for } n = 1, \dots, N \\
w_n &\sim \text{Multinomial}(z_n, \beta), \quad \text{for } n = 1, \dots, N
\end{aligned}
$$

Where z_n is the topic for word w_n. The dimensionality k of the Dirichlet distribution (i.e., the number of topics) is assumed to be known and fixed. The word probabilities are parametrized by the $k \times V$ matrix: β, where V is the vocabulary size. This matrix is treated as a fixed quantity that has to be estimated. [5]

Word2Vec

This is a package consisting of a collection of algorithms that produce semantic vector representations of words based on context words [6-7]. The most common ones are the skip-gram and the bag-of-words

algorithms, both of which are probabilistic classifiers that use simple neural networks to predict words. The latter tries to predict a target word based on its context, while the skip-gram, given a target word w and its context window of L words $c_{1:L}$, assigns a probability based on how similar this context window is to the target word. This probability is computed by applying the sigmoid function to the dot product of the embedded vectors of the words. These vectors are computed using gradient descent. To train the network correctly, we need to use both positive actual context words for the target w and negative examples, false context words for the target. To train the classifier, we use the cross-entropy loss.

$$\begin{aligned} L_{CE} &= -\log\left[P(+|w,c_p)\prod_{i=1}^{k}P(-|w,c_{n_i})\right] \\ &= -\left[\log\sigma(c_p\cdot w)+\sum_{i=1}^{k}\log\sigma(-c_{n_i}\cdot w)\right] \end{aligned}$$

Where $P(+|w,c_p)$ is the probability that the true context word vector c_p is a real context word of w, and $P(-|w,c_{n_i})$ is the probability that the noise word vector c_{n_i} isn't in the context of w. Usually more negative examples are use, to improve performance and avoid overfitting [6].

SentenceTransformers

This is also a collection of algorithms that produce semantic vector representations or embeddings, but in this case, for complete sentences rather than just words. In this case, state-of-the-art transformer models are used to predict a target word based on its context. Therefore, it performs the same prediction task as the bag-of-words algorithm, but this time using a much more complicated model. This package was originally developed to do an efficient semantic search between sentences (i.e., extract the most similar sentences to another sentence). This is explained in the Sentence-BERT (s-BERT) paper by the same authors as the package. The s-BERT uses the BERT (Bidirectional Encoder Representations from Transformers) model to compute sentence embeddings [8].

BERT is a state-of-the-art NLP model developed by Google, and it's broadly used in sentiment analysis. However, it's also useful for other tasks such as question answering, text generation, and text prediction. This model works by "masking" a word from a sentence and using the context both before and after this word (bidirectional) to predict it. Given that BERT is quite large and therefore computationally expensive, we didn't use it to encode our sentences. However, we used models that, although much smaller, are similar in principle [9].

Results

As mentioned previously, since this is an unsupervised learning task, the performance evaluation is mainly subjective. For the sentence embeddings approach, we used the class-based term frequency times inverse document frequency (TF-IDF) metric to analyze how distinct topics are from one another

[2]. To evaluate the LDA model, we used the computed beta probabilities per word per topic. We are still working on finding a way to measure the performance of the word embeddings approach. Since a label is assigned to each word in the vocabulary separately, we can't perform class-based TF-IDF. Additionally, we used boxplots to compare the distribution of hourly rates for the computed topics.

To select the clustering algorithm and its parameters in the vector semantic approach, we used the silhouette score. It was found that K-means with 10 clusters outperformed the Gaussian Mixture Model and DBSCAN. However, it's unclear how appropriate this metric is in this particular case, as it assumes that the clusters are mostly rounded with not too much noise.

Figure 6 shows the most common words that appear in a specific topic. As we can see words such as "children", "care" or "nanny" appear in almost all topics, this is often mentioned as an advantage of topic modelling with LDA, as opposed to hard-clustering methods, like the ones we tried for the word embeddings approach. However, in this case topics aren't distinguishable from one another.

For the sentence embeddings approach, a label is assigned to each sentence individually based on its vector representation. Therefore, even though we are using hard clustering, topics are allowed to share words as long as the context in which they are used isn't similar enough. In Figure 7, the values for the class-based TF-IDF are shown. Arguably, topics are more distinct with this approach. For example, topic 8 clearly refers to users talking about their rates, while topic 1 is all about forms of communication. In topic 3, the users talk about their qualifications or things they can provide. Topic 6 appears to be mostly adjectives, and topic 0 is represented by nannies talking about the ages of children they can take care of or have experience with.

In figure 8 we can see the difference in the distribution of requested hourly rates by topic. In this case it appears that the biggest differences are shown in the LDA model, this model is also distributing the users more uniformly among the topics, as no cluster has less than 150 users. On the other hand, the SentenceTransformer model has a cluster with 1389 users and another with 23, similar can be said about the word2vec approach, which has one cluster with 2099 users and three labels with less than 100.

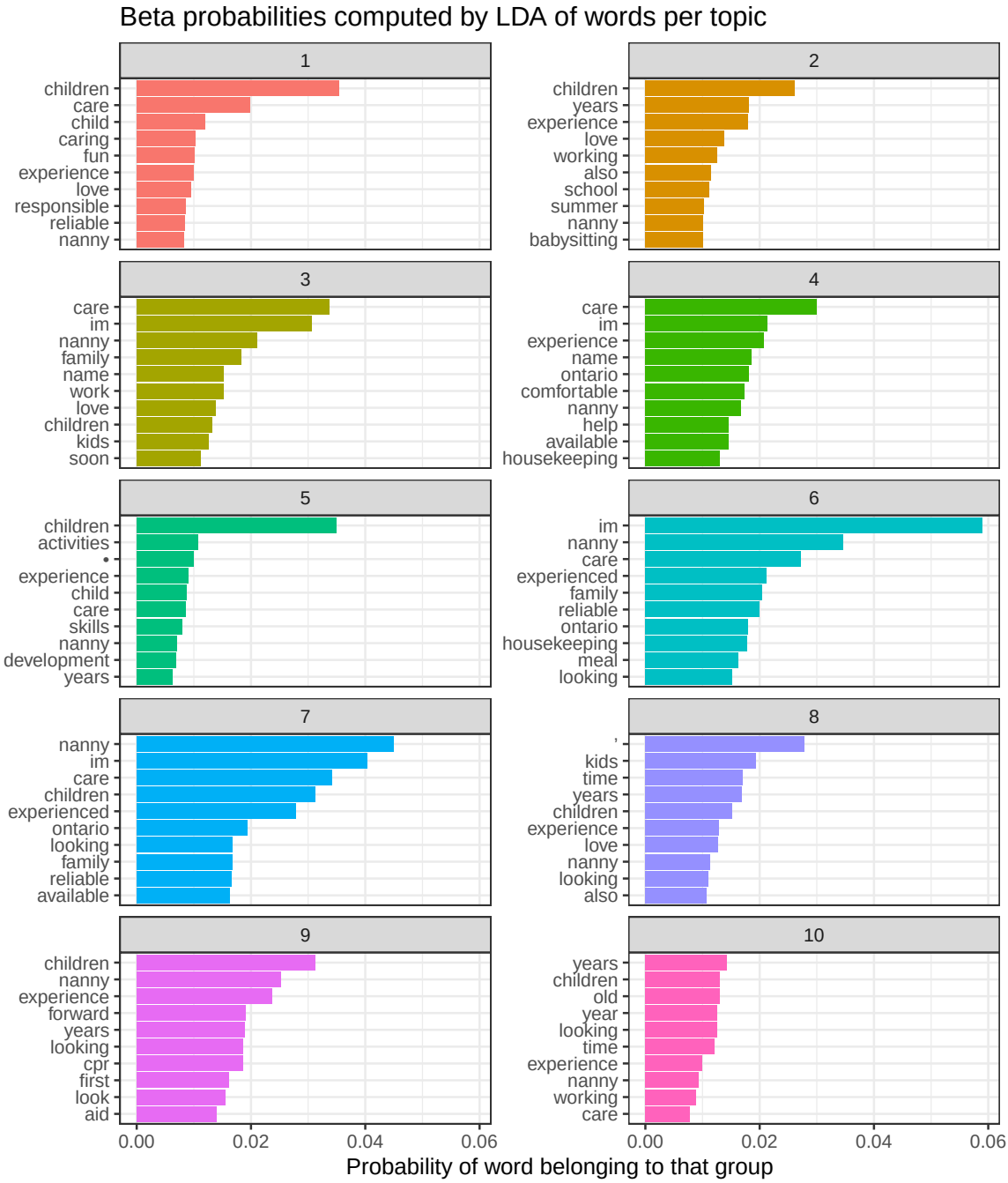


Figure 6: Resulting class based TF-IDF. Topics where selected for each observation using the highest probability of generation by the LDA model with 10 clusters

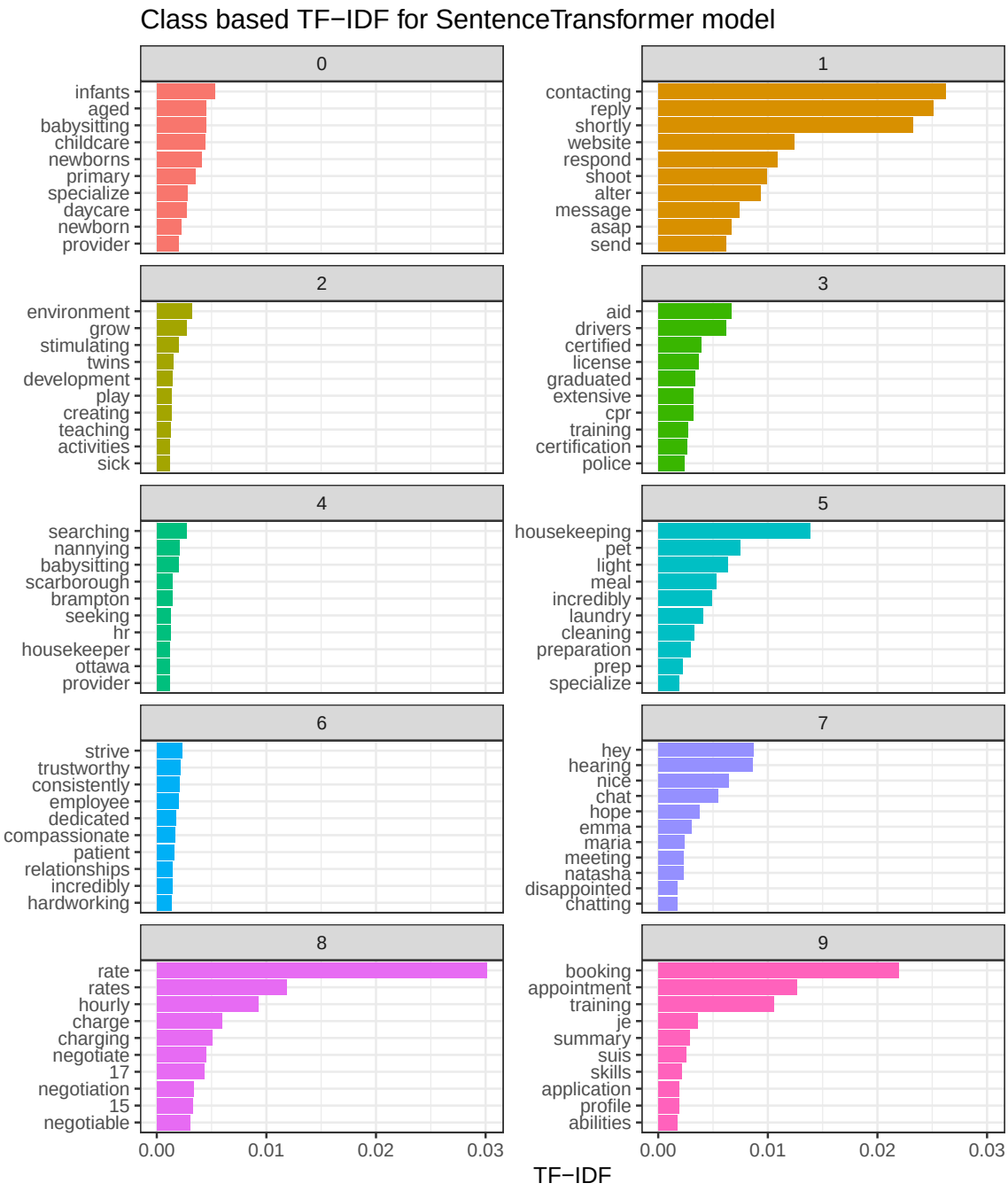
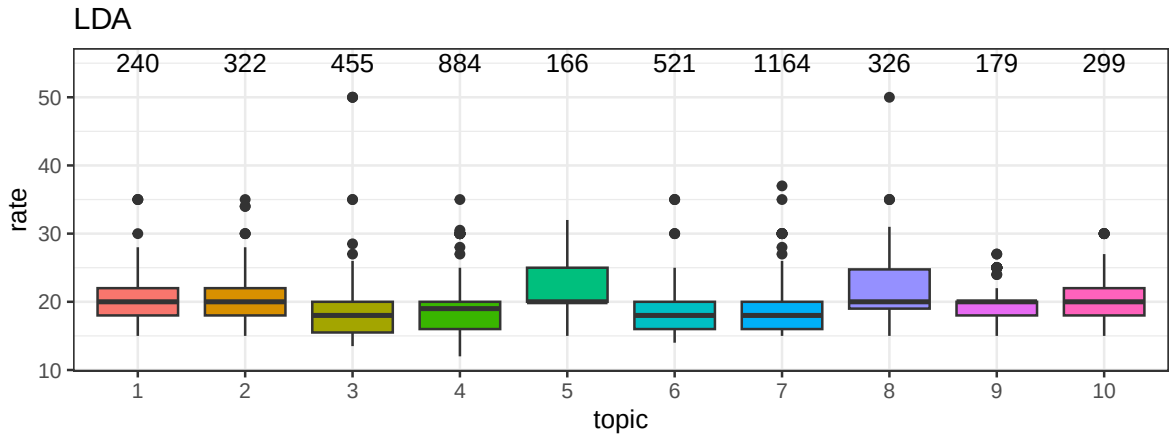
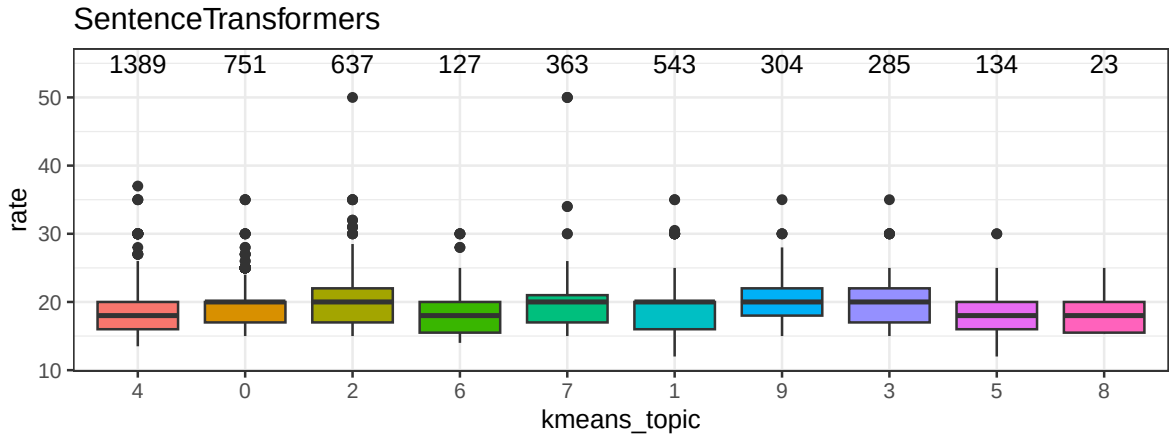
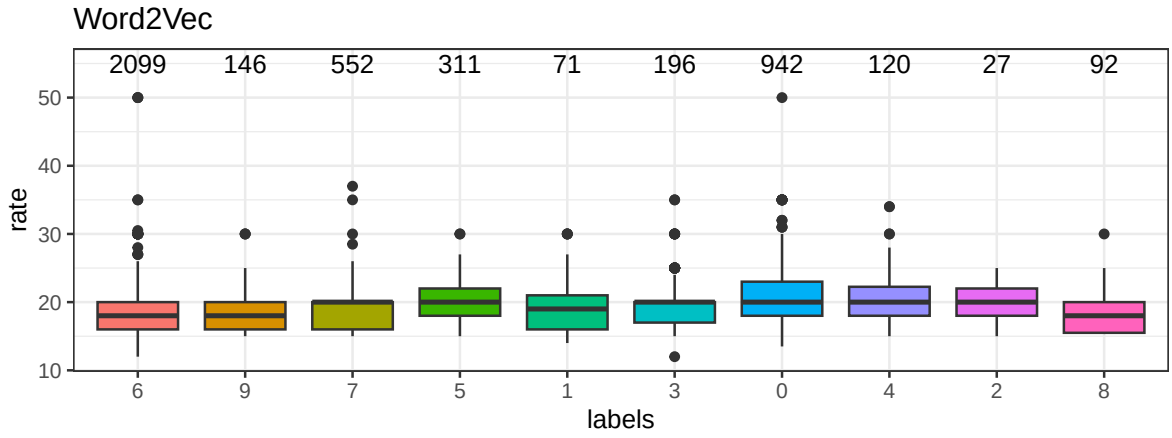


Figure 7: Resulting class based TF-IDF. Topics where selected for each sentence disregarding the user."


LDA
240
322
455
884
166
521
1164
326
179
299
rate
topic


SentenceTransformers
1389
751
637
127
363
543
304
285
134
23
rate
kmeans_topic


Word2Vec
2099
146
552
311
71
196
942
120
27
92
rate
labels

Conclusion

In this work, we analyze the profile descriptions of nannies across Ontario, which were scraped using R from a publicly available online website. From our analysis, it's clear that the users' profiles are very similar to one another, and most of them are based in Toronto. This posed a challenge as it was difficult to find paths to explore in more depth.

We tested a few different approaches for topic modeling. First, we used LDA to assign labels to each profile description. We also used BERT-generated sentence and skip-gram-generated word embeddings as input to a clustering algorithm. These approaches were quite difficult to compare as they are different in principle. For example, LDA sees documents (profile descriptions) as generated by topics, so it labels documents as a whole, while the semantic vector approach labels individual words or sentences. We used a class-based TF-IDF and found that the models generated using vector representation of sentences are better at distinguishing different parts of the text compared to the LDA model. However, the LDA model distributed users more uniformly into different topics. More than a measure of performance, this analysis shows the huge difference in the principles behind each approach. As mentioned before, both the SentenceTransformers and word2vec care only about sentences and words, respectively, and they completely disregard the document name (user name in this case), while LDA is specifically trying to separate individual documents into clusters. However, it isn't very clear what is distinct about the clusters generated by LDA.

All these models have their advantages, depending on the task or goal we are interested in. For example, if we have a corpus of chapters from books of different genres and we want to predict which chapter was written by which author, then LDA will give excellent performance [2]. While if we want to do a semantic sentence search (i.e., which sentence in a specific text is the most similar to a target sentence), then the SentenceTransformer approach will give way better results [8]. For the future, we are interested in improving our testing and data retrieval pipelines, as well as establishing a more objective way of measuring the performance of the models.

References

[1] CareGuide: CanadianNanny.ca. Retrieved July 3, 2023 from: https://canadiannanny.ca/nannies/ontario.

[2] Silge, J., &; Robinson, D. (2017). Text mining with R: A tidy approach. O'Reilly.

[3] Pedersen T (2022). ggraph: An Implementation of Grammar of Graphics for Graphs and Networks. https://ggraph.data-imaginist.com, https://github.com/thomasp85/ggraph.

[4] Csardi G, Nepusz T (2006). "The igraph software package for complex network research." InterJournal, Complex Systems, 1695. https://igraph.org.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3, null (3/1/2003), 993–1022.

[6] Jurafsky, D., & Martin, J. H. (2022). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Pearson.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, & Jeffrey Dean. (2013). Efficient Estimation of Word Representations in Vector Space.

[8] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.