User-defined Content Detection Framework

Keonwoo Kim[1,2] **Sungzoon Cho**[1] **Younggun Lee**[2*]
[1]Seoul National University [2]Neosapience
{keonwookim@dm,zoon@}snu.ac.kr
yg@neosapience.com

Abstract

This paper presents UCDF, a simple interactive framework for training a classifier that can detect user-defined content. User-defined content signifies texts that a user wants to detect. UCDF receives examples of user-defined content as queries to make a customized dataset automatically using a similar method to dense passage retrieval. UCDF uses the customized dataset to train a classifier, a user-defined content detector. We experimented with a hate speech detection task by setting input queries related to the task, and it shows a competitive level of performance to existing models for toxic detection. We also tested UCDF with arbitrary user-defined content detection tasks to check the classification performance of UCDF. Our code is available at https://github.com/UCDFInterNLP/UCDF.

Warning: This paper used samples of hate speech text.

1 Introduction

As social media became prevalent, hate speech that forms readers' displeasure is emerging as a significant social problem. Even chatbots, such as Tay from Microsoft, have caused social controversy by generating toxic sentences. To detect text containing hate expressions generated by artificial intelligence or humans, various studies related to hate speech detection using deep learning models have been conducted (Hartvigsen et al. [2022], Röttger et al. [2020], Aluru et al. [2020], Alkomah and Ma [2022]). However, they are mainly focused on classifying general toxic texts such as offensive remarks about gender, race, and religion. It can be more beneficial to have a generalized version of a hate speech detector, such as a user-defined content detector. For instance, company X may want to filter out content about their rival from company X's bulletin board. A naive approach for this task is to build a labeled dataset to classify the specific content and then train a classifier on it. The problem with this approach is that it is costly to build such a dataset to train a user-defined content detector for each user. In this paper, we first introduce the user-defined content detection task as a new research topic and propose a novel interactive framework called the user-defined content detection framework (UCDF) to solve this task. UCDF utilizes dense passage retrieval (DPR, Karpukhin et al. [2020]) to easily build a customized dataset by receiving examples of user-defined content as queries from a user. Then train a classifier with the customized dataset. For quantitative comparison, we applied UCDF to the hate speech detection task, which has a labeled dataset and existing models with the defined queries. In addition, we also experimented with user-defined content detection tasks using arbitrary user-defined contents to determine whether UCDF performs well on user-defined content detection tasks.

The contributions of our paper are as follows:

- We present a user-defined content detection task as a new research topic.
- We propose a simple interactive framework detecting text related to user-defined content.
- We suggest a novel method to build a customized dataset for the detection task.

*Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

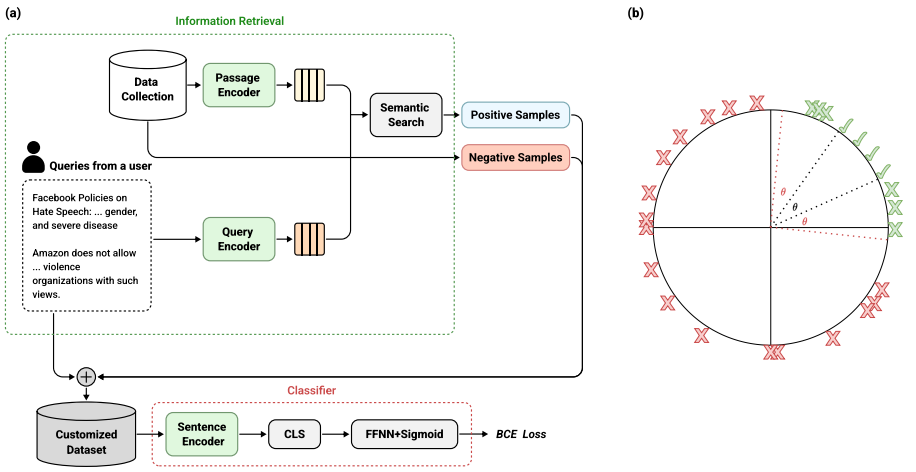

Figure 1: (a) Overall pipeline of UCDF. (b) A circle represents the normalized sentence embedding of all sentences. ✔denotes examples of user-defined content as queries, and ✖denotes passages in data collection. θ indicates the cosine of an angle of two input queries with the lowest cosine similarity. For each query, passages within an angle size θ are retrieved as positive samples.

2 Related work

Information retrieval Information retrieval is the task of finding semantically similar documents for a query. With the recent development of deep learning, studies that retrieve documents related to queries based on dense embeddings, such as DPR (Karpukhin et al. [2020]) and TAS-Balanced (Hofstätter et al. [2021]), showed better performance than sparse-based methods such as BM25 (Robertson et al. [2009]). DPR, trained using dual-encoders with metric learning, focuses on retrieving passages contextually related to question queries in QA tasks. It works by indexing all documents into a continuous space and retrieving N passages semantically related to the question query.

3 Method

Figure 1(a) illustrates the process of UCDF. UCDF includes two stages: (1) building a customized dataset from the queries using DPR; (2) training a classifier using the customized dataset. In the first stage, queries and data collection are required to apply DPR. We prepare example sentences or keywords of the user-defined content to form queries. The data collection is a corpus of indexed passages that can cover various domains. We retrieve passages from the data collection according to the queries to build positive samples, and negative samples are randomly sampled passages. The queries and the passage encoder are used for retrieval as in (Karpukhin et al. [2020]). The detailed process of the retrieval is described in Section 3.1. In the second stage, we fine-tune a pretrained sentence encoder on the binary classification task with the customized dataset. The fine-tuning process is similar to (Devlin et al. [2018]) that feeds a classification token ([CLS]) to a feedforward neural network (FFNN). It uses binary cross-entropy loss as an objective function. In this way, we can obtain a user-defined content detector. An additional interactive stage can be placed after the training of the classifier. We add more queries to refine the customized dataset to train a better user-defined content detector. At this stage, we may add negative queries of the opponents of the user-defined content. Negative samples formed by the negative queries can help the classifier better discern the user-defined content.

3.1 Customized dataset

A visualization of constructing positive samples can be seen in Figure 1(b). We collect N example sentences or keywords of user-defined content from a user to form N queries. While building positive samples, the minimum or the average value of pairwise cosine similarities between the queries is used as a threshold. Only passages whose cosine similarity score exceeds the threshold for each

Table 1: Classification results on hate speech dataset. () refers to threshold criteria while building a customized dataset. The encoder types used for dual and sentence encoders are the same.

	Tweets Hate Speech Detection			Davidson et al.		
Model	F1-score (%)	Accuracy (%)	AUC (%)	F1-score (%)	Accuracy (%)	AUC (%)
RoBERTa [a][2]	19.94	90.96	67.13	**37.61**	46.41	**63.52**
RoBERTa [b][3]	19.95	90.21	23.35	34.84	45.80	59.31
UCDF - SimCSE (avg)	**32.06**	**91.65**	78.47	13.02	**77.72**	22.21
UCDF - SimCSE (min)	30.54	79.51	**81.13**	23.11	74.84	58.38

query vector are retrieved among the passage vectors in the data collection, and M positive samples are formed. A total of $\mathrm{N}+\mathrm{M}$ positive samples are constructed by combining N queries and M passages. After randomly retrieving passages for negative samples of the same size as the positive samples from data collection, we finally construct a customized dataset for training a classifier with a total of $2(\mathrm{N}+\mathrm{M})$ samples. If the user wants to enter additional negative queries to improve UCDF performance, the method of retrieving passages for negative samples is the same as in positive samples.

4 Experiments and results

All the encoders used in UCDF are initialized with pretrained sentence encoders that have the same weight, but the weights were not shared during the training of UCDF. We used SimCSE (Gao et al. [2021]), state-of-the-art sentence embedding method, for the pretrained sentence encoders. While using SimCSE, we manually trained encoders on the NaturalQuestion dataset from scratch. For passages in the data collection, we used the same passages used in Meta's open-source repository (Facebookresearch [2020]), from which we employed the pretrained encoder.

4.1 Hate speech detection

Experimental setup Since user-defined content detection task has never been studied before, there is no baseline model for directly evaluating UCDF. Using the attribute of UCDF that can receive arbitrary queries, we quantitatively analyzed UCDF by applying it to hate speech detection tasks. We used the protected characteristics sentences specified by big tech companies as queries. Examples of the queries can be found in Appendix A.2, and negative queries are not used in this experiment. Baseline models are two RoBERTa-based (Liu et al. [2019]) models, fine-tuned on the hate speech detection tasks. The first model[2] was trained on Jigsaw dataset with approximately 2 million samples. The second model[3] was trained on machine-generated toxic texts. For a fair comparison, we selected two test datasets that the UCDF and the baseline models had never seen before.

Test data The hate speech datasets used to verify UCDF's performance are as follows.

1. Tweets hate speech detection (Sharmaroshan [2019]): A dataset that is composed of tweets on Twitter. It consists mainly of standardized tweets without abusive expressions.
2. Davidson et al.: A dataset from (Davidson et al. [2017]) that is related to the hate speech detection task. Since the dataset consists of three label values ['hate speech', 'offensive language', and 'those with neither'], we removed 'offensive language' for binary classification. It consists mainly of rough slang or swear words.

Results analysis We evaluated the baseline models and the UCDF through unseen datasets. We reported the F1 score, Accuracy, and AUC for each setting. In the case of the Tweets Hate Speech Detection dataset, Table 1 shows that the performance of UCDF with SimCSE encoders outperforms the baseline models. However, it shows lower performance than RoBERTa for Davidson et al. dataset in terms of F1 score. Note that the UCDF's customized dataset is constructed from Wikipedia, consisting mainly of standardized expressions, whereas the Davidson et al. dataset mainly consists of

[2]https://huggingface.co/SkolkovoInstitute/roberta_toxicity_classifier
[3]https://huggingface.co/tomh/toxigen_hatebert

Table 2: Classification results on the customized dataset. Three methods to build negative samples of a customized dataset. ('rand': random sampling, 'nq': sampling based on negative queries)

	Customized Religion			Customized South Korea		
Sampling method	F1-score (%)	Accuracy (%)	AUC (%)	F1-score (%)	Accuracy (%)	AUC (%)
w/o nq, w/ rand	78.95	79.49	84.69	78.10	79.46	84.18
w/ nq, w/o rand	68.97	76.92	82.70	72.58	69.64	76.79
w/ nq, w/ rand	**95.24**	**96.15**	**97.53**	**85.44**	**86.61**	**93.94**

rough abusive language. It made UCDF difficult to learn toxic patterns from the customized dataset. Considering only the Tweets Hate speech detection dataset, it is better to set the threshold type to average than the minimum.

4.2 User-defined content detection

Experimental setup In this experiment, we used UCDF with threshold type as an average value, accordingly with the results in Section 4.1. We set two arbitrary user-defined contents to experiment on UCDF. Since there is no test data for the arbitrary contents, we built two test data for this experiment. Every sentence in the test data is labeled with either positive or negative. Note that the negative samples include hard negatives and easy negatives. Queries used in this experiment can be found in Appendix A.3. When constructing a customized dataset, we set up negative samples in three methods. The first method is to sample from data collection randomly. The second is to receive negative queries from the user and extract passages in the same way as in Figure 1(b). The third is combination of the first and the second methods. If there were overlaps between positive and negative samples, we delete the overlapped samples. We reported the F1 score, accuracy, and AUC for each setting.

Test data Dataset for user-defined content detection to verify UCDF's performance are as follows.

1. Customized religion dataset[4]: Positive samples are related to Catholicism, Christianity, and Islam. Easy negative samples are independent of religion, and hard negative samples are within the category of religions such as Buddhism and Hinduism.
2. Customized South Korea dataset[5]: Positive samples are related to South Korea. Easy negative samples are independent of South Korea, and hard negative samples are within the category of the country but are not related to South Korea. In addition, we added content related to Northeast Asia, such as North Korea, China, and Japan, in hard negative samples.

Results analysis Table 2 shows the evaluation results on the user-defined content detection task. As shown in Appendix A.3, building negative samples of a customized dataset by the first method made UCDF not to detect the hard negatives well, since the hard negative samples and the positive samples both contain semantically similar content. Also, constructing negative samples by the second method made UCDF correctly detect the hard negatives, but not the easy negatives. The second method aroused bias for the negative queries in the fine-tuning process. On the other hand, setting negative samples with the third method made UCDF significantly outperform other cases in all metrics. This is because the customized dataset is built to complement the bias for negative queries and classify general contents independent of positive samples.

5 Conclusion

In this work, we defined a novel research topic that detects user-defined content detection, and propose UCDF as a solution. In the future, we may improve the classification performance of UCDF by replacing the data collection with better dataset such as the Common Crawl dataset, which contains more diverse contents and expressions than Wikipedia. One limitation of our work is that the data collection in UCDF must be continuously updated to detect the latest content. We should be

[4]https://www.history.com/topics/religion
[5]https://facts.net/south-korea-facts/

concerned about issues related to information privacy that may arise from UCDF. Since it interactively receives queries as input from users, private information about a specific individual may be leaked.

References

Fatimah Alkomah and Xiaogang Ma. A literature review of textual hate speech detection methods and datasets. *Information*, 13(6):273, 2022.

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*, 2020.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Facebookresearch. Dense passage retrieval, 2020. URL https://github.com/facebookresearch/DPR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Paul Röttger, Bertram Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet B Pierrehumbert. Hatecheck: Functional tests for hate speech detection models. *arXiv preprint arXiv:2012.15606*, 2020.

Sharmaroshan. Twitter-sentiment-analysis, May 2019. URL https://github.com/sharmaroshan/Twitter-Sentiment-Analysis.

A Appendix

A.1 Training details

We use three V100 GPUs for training SimCSE-based dual-encoders on the NaturalQuestion dataset. The average running time and configuration are similar to DPR, BERT-based dual-encoders. While fine-tuning a user-defined detector, we set the maximum length in the number of tokens to 128 and the batch size to 32. Also, we use AdamW(Loshchilov and Hutter [2017]) with 5e-5 of learning rate.

A.2 UCDF on hate speech detection experiment

Table 3: Details in building positive samples of a customized dataset on hate speech detection task.

	Threshold types		
	avg	min	max
Threshold value	0.8473	0.7658	0.9166
# of retrieved positive samples	560	22,037	3

Table 3 shows the number of positive samples built along the type of threshold used to build a customized dataset. If the threshold is set to the maximum value, we do not experiment due to the small size of the customized dataset.

Table 4: Examples of queries used in hate speech detection task

Company name	**Query**
Facebook	Facebook Policies on Hate Speech: We define hate speech as a direct attack against people — rather than concepts or institutions— based on protected characteristics: race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity, and severe disease.
Youtube	We remove content promoting violence or hatred against individuals or groups based on any of the following attributes: race or ethnic origin, religion, disability, gender, age, veteran status, or sexual orientation/gender identity.
Twitter	You may not promote violence against or directly attack or threaten other people based on race, ethnicity, national origin, caste, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease.
Vimeo	We do not allow hateful and discriminatory speech. We define this as any expression that is directed to an individual or group of individuals based upon the personal characteristics of that individual or group. Outside of content, Vimeo also monitors avatars, screen names, and profile pictures for hate speech.
Amazon	Amazon does not allow products that promote, incite, or glorify hatred, violence, racial, sexual, or religious intolerance or promote organizations with such views.
Snapchat	Hate speech or content that demeans, defames, or promotes discrimination or violence on the basis of race, color, caste, ethnicity, national origin, religion, sexual orientation, gender identity, disability, or veteran status, immigration status, socio-economic status, age, weight or pregnancy status is prohibited.
Spotify	Don't engage in any activity, post any User Content, or register or use a username, which is or includes material that is offensive, abusive, defamatory, pornographic, threatening, or obscene, or advocates or incites violence.

Table 4 shows the examples of queries for UCDF used in the hate speech detection task. To get high-quality results, we prepare official queries[6] from seven well-known companies.

A.3 UCDF on user-defined content experiment

Table 5: Examples of queries used in user-defined content detection experiment (Religion)

Class	Query
Positive	Judaism is the world's oldest monotheistic religion, dating back nearly 4,000 years.
Positive	Christianity is the most widely practiced religion in the world, with more than 2 billion followers.
Positive	Islam is the second largest religion in the world after Christianity, with about 1.8 billion Muslims worldwide.
Negative	Buddhism is a faith that was founded by Siddhartha Gautama ("the Buddha") more than 2,500 years ago in India.
Negative	Hinduism is the world's oldest religion, according to many scholars, with roots and customs dating back more than 4,000 years.

Table 6: Examples of queries used in user-defined content detection experiment (South Korea)

Class	Query
Positive	South Korea, K-pop, Seoul, Kimchi
Negative	North Korea, Japan, China

Table 5 and Table 6 shows examples of queries for UCDF conducted in experiment 4.2. In the religion content detection experiment, UCDF received queries describing each religion (Judaism, Christianity, Islam, Buddhism, Hinduism). On the other hand, in the case of the South Korea content detection experiment, UCDF received the queries as a keyword. As shown in 2, we find that UCDF works well regardless of whether the query is a sentence or a keyword. In addition, since UCDF encodes input queries with [CLS] token, we can also enter document-level texts as queries.

Test data We build test data containing 78 samples for religion content detection. Test data contains binary labels with positive and negative. Hard and easy negatives are classified as the same class in the test data. The number of positive, hard negative, and easy negative samples is 31, 25, and 22, respectively. In the same way as building test data related to the religious content, test data containing 112 samples for South Korea content detection is constructed. The number of Positive, hard negative and easy negative samples is 56, 34, and 22, respectively. The easy negative samples used in the two test data are the same, independent of religion and South Korea. We disclose built test data[7].

[6] https://blog.ongig.com/diversity-and-inclusion/human-rights-policy-hate-speech-policy
[7] https://github.com/UCDFInterNLP/UCDF

Table 7: Accuracy score by each class. ('rand': random sampling, 'nq': sampling based on negative queries)

	Customized Religion			Customized South Korea		
Sampling method	Positive (%)	Easy Negative (%)	Hard Negative (%)	Positive (%)	Easy Negative (%)	Hard Negative (%)
w/o nq, w/ rand	**96.77**	**100.00**	40.00	73.21	**100.00**	76.47
w/ nq, w/o rand	64.52	86.36	84.00	**80.36**	18.18	85.29
w/ nq, w/ rand	**96.77**	**100.00**	**92.00**	78.57	**100.00**	**91.18**

As shown in Table 7, in most cases, the performance is the best when the random sampling method and the sampling based on negative queries are applied for building negative samples, among other sampling methods. It shows that the performance of UCDF significantly depends on the method of building negative samples.

Table 8: Examples of test data in user-defined content detection experiment (Religion)

Class	**Example**
Positive	Jewish people believe there's only one God who has established a covenant—or special agreement—with them.
Positive	Muslims are monotheistic and worship one, all-knowing God, who in Arabic is known as Allah.
Easy Negative	Enter your destination & your Tesla will automatically include Supercharging stops in your route
Easy Negative	Comments section of Yahoo controlled by alt-reality biased moderators supporting lies harmful to the Nation.
Hard Negative	The religion's founder, Buddha, is considered an extraordinary being, but not a god. The word Buddha means "enlightened."
Hard Negative	Hinduism is unique in that it's not a single religion but a compilation of many traditions and philosophies.

Table 9: Examples of test data in user-defined content detection experiment (South Korea)

Class	**Example**
Positive	Bong Joon Ho's Parasite made history for bagging 3 awards at the 2020 Oscars, which was the most of any film nominated.
Positive	Hangul classifies as one of the Altaic languages, is affiliated to Japanese, and contains some Chinese loanwords.
Easy Negative	Recently after more than 20 years as a Google account holder my YouTube channel was suspended without warning and without any reason given.
Easy Negative	Jordi Cruyff has signed his contract as FC Barcelona's new sporting director of football. He has already been serving in the role since July 1.
Hard Negative	The greatest health threat in North Korea is hunger.
Hard Negative	The Huizhou Ancient Town is a famous historical and cultural city in southern Anhui Province with over 2000 years of history.

Examples of test data in user-defined content detection experiment are presented in Table 8 and Table 9. We prepared two examples that UCDF correctly classified for each class. Hard Negative consists of examples difficult for people to classify without background knowledge of the content.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Introduction 1
   (b) Did you describe the limitations of your work? [Yes] See Conclusion 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Conclusion 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code is provided in a URL.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Provided code contains details. Also, there are not critically important hyperparameters in this experiment.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.1
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Reference.
   (b) Did you mention the license of the assets? [Yes] See Reference and Method 3.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included ULR of the code in the paper
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See section A.3.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] manually on a sample. Personally identifiable information cannot be identified in the dataset.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]