Exploring Universal Sentence Encoders for Zero-shot Text Classification

Souvika Sarkar, Dongji Feng, Shubhra Kanti Karmaker Santu
Big Data Intelligence (BDI) Lab, Department of Computer Science & Software Engineering
Auburn University, Alabama, USA
{szs0239, dzf0023, sks0086}@auburn.edu

Abstract

Universal Sentence Encoder (USE) has gained much popularity recently as a general-purpose sentence encoding technique. As the name suggests, *USE* is designed to be fairly general and has indeed been shown to achieve superior performances for many downstream NLP tasks. In this paper, we present an interesting "negative" result on *USE* in the context of *zero-shot* text classification, a challenging task, which has recently gained much attraction. More specifically, we found some interesting cases of *zero-shot* classification, where topic based inference outperformed *USE*-based inference in terms of F_1 score. Further investigation revealed that *USE* struggles to perform well on datasets with a large number of labels with high semantic overlaps, while topic-based classification works well for the same.

1 Introduction

What makes a sentence encoder *universal*? The tantalizing idea is to learn a general sentence encoding technique that can achieve "good" performance on a wide variety of downstream tasks. Recently, Google's *Universal Sentence Encoder* (USE) Cer et al. (2018) has been shown to achieve great success in various downstream tasks and promising results in a way provided some justification to the name "Universal Sentence Encoder" itself.

While *USE* Cer et al. (2018) is undoubtedly one of the state-of-the-art sentence encoding techniques available today, it's success has primarily been demonstrated within the "pre-train/fine-tune" paradigm, where, it is assumed that the target labels are known beforehand as well as a small amount of training data is readily available, which can facilitate the fine-tuning process. Whereas, a more challenging task is zero-shot text classification Yin et al. (2019), where, neither the target labels are known beforehand nor any training data is available for fine-tuning. How *USE* performs in case of zero-shot text classification is, therefore an interesting research question, which is relatively underexplored at this moment.

To address this knowledge gap, we performed a systematic study, where, we applied *USE* to perform the "Zero-shot Text Classification" task, as defined by Yin et al. (2019). The goal of our study is to investigate how powerful *USE* is for solving an NLP task for which acquiring training data is almost impractical.

To perform this study, we conducted extensive experiments with seven real-world datasets. As a baseline, we implemented two topic-based zero-shot classification techniques for comparative analysis. We evaluated the goal-task performance against the "Gold" standard labels annotated by humans and computed F_1 metric for each method compared. Experimental results demonstrate that topic-based inference clearly outperformed *USE*-based inference in terms of F_1 score for most of the datasets, essentially yielding the so-called "negative" result. Further investigation revealed that *USE* struggles to perform well on datasets with a large number of labels with high semantic overlaps, while topic-based methods work well for the same.

2 Background and Related Work

Universal Sentence Encoder: The utility of *USE* has been tested for many popular NLP tasks including *Intent Classification* Casanueva et al. (2020), *Fake-News Detection* Majumder and Das (2020), *Duplicate Record Identification* Lattar et al. (2020) and *COVID-19 Trending Topics Detection* from tweets Asgari-Chenaghlu et al. (2020). Perone et al. (2018); Enayet and Sukthankar (2020) focused on the performances of different sentence embedding techniques for transfer-learning tasks. Rivas and Zimmermann (2019) reported that state-of-the-art sentence embeddings are unable to capture sufficient information regarding sentence correctness and quality in the English language.

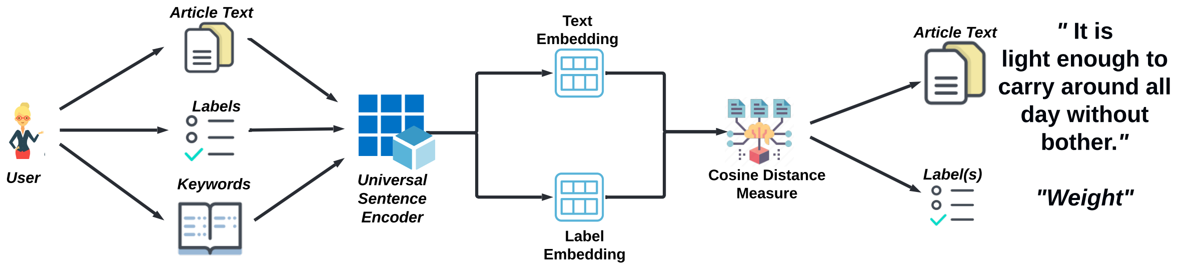


Figure 1: Steps for Zero-shot Text Classification leveraging Universal Sentence Encoder.

Zero-Shot Classification: Veeranna et al. (2016) adopted pre-trained word embedding for measuring semantic similarity between a label and documents. (Hascoet et al., 2019; Zhang et al., 2019; Xie and Virtanen, 2021), performed zero-shot learning using semantic embedding. Rios and Kavuluru (2018) attempted to understand how state-of-the-art topic inference methods perform on infrequent labels. Rios and Kavuluru (2018) explored ***few-shot*** and ***zero-shot*** learning methods for multi-label text classification. Yin et al. (2019) established a benchmark for zero-shot text classification problem by providing unified datasets, standardized evaluations. Xia et al. (2018) studied the zero-shot intent detection problem for detecting user intents without any labeled utterances. Pushp and Srivastava (2017) proposed "TRAIN ONCE, TEST ANYWHERE" approach which involves training model to tackle unseen sentences, tags, and new datasets. Puri and Catanzaro (2019) proposed generative models for zero-shot text classification. Recently, Chen et al. (2021) implemented zero-shot text classification via Knowledge Graph Embedding for Social Media Data. Gong and Eldardiry (2021) discussed about zero-shot learning's settings, methods, and applications.

Uniqueness of This Work: We explore the efficacy of *USE* for "Zero-shot Text Classification" task and compare against topic-based zero-shot methods, which is unique about this work.

3 Zero-shot Text Classification

Zero-shot Text Classification (0SHOT-TC) is a challenging problem which aims to associate an appropriate label with a piece of text, regardless of the text domain without any training/fine-tuning. The idea of zero-shot TC was coined by Yin et al. (2019), and in this paper, we have specifically focused on Definition-Wild 0SHOT-TC discussed by Yin et al. (2019), a visual depiction of which is presented in Figure 1. More specifically, we formalize our task as below:

Definition 1. ***0SHOT-TC:*** *Given a collection of text articles $T = \{t_1, t_2, ..., t_n\}$, a user x and a set of* ***user-defined*** *labels $L_x = \{l_1, l_2, ..., l_m\}$ provided in* ***real-time****, classify each text article $t_i \in T$ with zero or more labels from L_x* ***without any further fine-tuning****.*

Notably, it is possible that two different users will focus on different set of labels for the same dataset based on their application needs. Furthermore, creating customized training datasets beforehand is no longer possible because the target labels are provided in real-time by users.

3.1 *USE* Based Zero-shot Text Classification

The steps to classify text using Universal Sentence Encoder is discussed in algorithm 1 and shown in Figure 1. We used both DAN[1] and Transformer[2] based *USE* models Cer et al. (2018) to encode target-labels and the article-text. Next, based on the cosine similarity score between a label-embedding and the article text-embedding, the particular label is assigned if the similarity is higher than a threshold, or dropped otherwise.

Algorithm 1 Zero-shot TC using sentence encoder

```
Input: Article text, Labels and Keywords
Output: Articles labeled with zero to many labels
Article text and label are converted into Text and Label embeddings using Universal Sentence Encoder
Measure cosine similarity between Text and Label embeddings
for $threshold = 0.0, 0.05, \ldots, 1$ do
    if cosine similarity > threshold then
        classify text with label
    end if
end for
```

[1] https://tfhub.dev/google/universal-sentence-encoder/4

[2] https://tfhub.dev/google/universal-sentence-encoder-large/5

Also, we adopted two different ways for target label embedding: 1) Label embedding using article-text which contains explicit mentions of label names (P1) and 2) Label embedding using label name and keywords (P2). The details of these embeddings have been discussed in appendix A.2.1 and A.2.2, respectively.

4 Experimental Design

4.1 Datasets for Case-Study

In our experiments we worked with 2 different type of datasets. (A) Large datasets (Medical and News datasets) having article count > 2000 and average article length as 641, collected from Sarkar and Karmaker (2022), and (B) Small datasets (User review datasets: Cellular phone, Digital camera1, Digital camera2, DVD player, Mp3 player) having article count < 2000 and average article length as 17, created by Hu and Liu (2004) and annotated by Karmaker Santu et al. (2016). Some statistics about these datasets are presented in Table 1, whereas details such as label names, label count, keywords etc. had been discussed on the respective papers. Both the datasets are already tagged with one or more labels (ground truth) and also each label is defined by a set of respective informative keywords. The keywords serves the purpose of auxiliary information Akata et al. (2016), required to perform zero-shot classification tasks (more details in Appendix A.1).

Dataset	Articles	# of Labels	Labels/article
Medical	2066	18	1.128
News	8940	12	0.805
Cellular phone	587	23	1.058
Digital camera1	642	24	1.069
Digital camera2	380	20	1.039
DVD player	839	23	0.781
Mp3 player	1811	21	0.956

Table 1: Statistics on large and small datasets

4.2 Methods, Baseline and Evaluation

As our baseline, we implemented a constrained topic-based zero-shot classification approach (based on the **Generative Feature Language Models** (GFLM) proposed by Karmaker Santu et al. (2016)). More specifically, we implemented two variants of the baseline approach: 1) GFLM-S (inference based on topic distribution of an entire document) and GFLM-W (inference based on topic distribution of a single word). This approach is based on generative probabilistic model which is a unsupervised statistical learning. The parameters are optimized automatically using an Expectation-Maximization algorithm in an unsupervised fashion; hence no training is required and consequently, can be considered as zero-shot [for details, see Karmaker Santu et al. (2016)]. For *USE*, we implemented four different *Zero-shot Text Classifiers*: 1) *USE* with Transformer architecture and P1 label embeddings (USE_T^{P1}). 2) *USE* with Transformer architecture and P2 label embeddings (USE_T^{P2}). 3) *USE* with DAN architecture and P1 label embeddings (USE_D^{P1}). 4) *USE* with DAN architecture and P2 label embeddings (USE_D^{P2}). As evaluation metric, we report the traditional *Precision*, *Recall* and the F_1 scores. To compute the F1 score, we first sum the respective True Positive, False Positive, and False Negative values across all labels and then plug them into the F1 equation to get micro-averaged F1 score.

5 Results and Findings

We first present the results on the seven datasets used in our experiments for the four variants of the *USE*-based *Zero-shot Text Classifier*s. Table 2 summarizes performance of the classifiers, which demonstrated that DAN based architectures performed slightly better than the transformer based architecture overall, while P1 label embeddings turned out to be superior than the P2 embeddings.

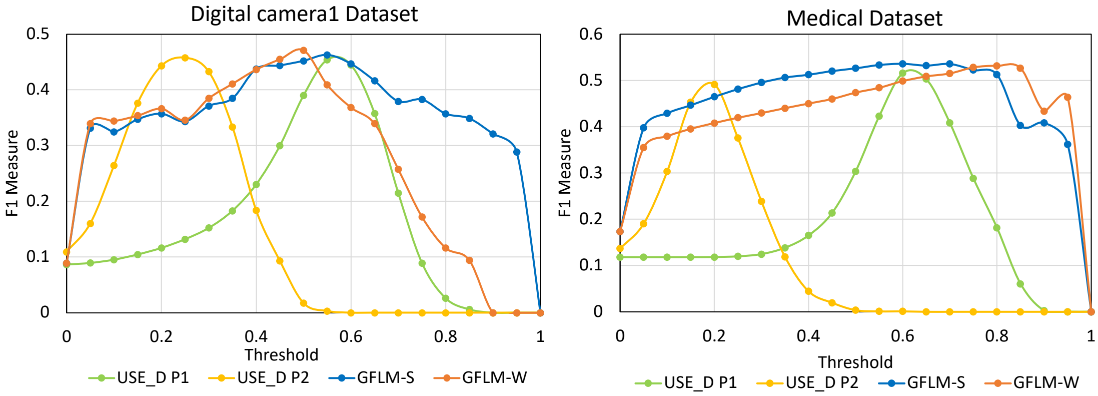
Dataset	$\text{USE}_\text{T}^\text{P1}$	$\text{USE}_\text{T}^\text{P2}$	$\text{USE}_\text{D}^\text{P1}$	$\text{USE}_\text{D}^\text{P2}$
Medical	0.503	0.486	**0.516**	0.495
News	0.438	0.423	0.445	**0.464**
Cellular phone	**0.486**	0.484	0.483	0.482
Digital camera1	0.408	0.447	**0.457**	0.454
Digital camera2	0.438	**0.505**	0.501	0.483
DVD player	**0.449**	0.403	**0.449**	0.440
Mp3 player	0.463	0.391	**0.466**	0.401

Table 2: F_1 Measure for *USE*-based classifiers with different embeddings. P1 denotes *Label embedding using explicit annotated text* and P2 denotes *Label embedding using label name and keywords.*

Based on the findings above, we further looked into the *precision* and *recall* scores of the DAN-architecture based *USE* classifiers (reported in Table 3) along with the baseline methods, GFLM-W and GFLM-S. It is evident from Table 3 that GFLM-W and GFLM-S perform significantly better than *USE* in terms of *precision*. Although in some cases, *recall* values of *USE* approaches were found to be better than the GFLM-W and GFLM-S, one should

	$\text{USE}_{\text{D}}^{\text{P1}}$			$\text{USE}_{\text{D}}^{\text{P2}}$			GFLM-S			GFLM-W		
Dataset	**Precision**	**Recall**	$\mathbf{F_1}$	**Precision**	**Recall**	$\mathbf{F_1}$	**Precision**	**Recall**	$\mathbf{F_1}$	**Precision**	**Recall**	$\mathbf{F_1}$
Medical	0.447	0.611	0.516	0.475	0.517	0.495	0.597	0.481	**0.533**	0.597	0.477	0.530
News	0.437	0.445	0.445	0.400	0.550	0.464	0.564	0.440	**0.494**	0.562	0.437	0.492
Cellular phone	0.398	0.612	0.483	0.407	0.594	0.482	0.494	0.501	0.498	0.480	0.529	**0.504**
Digital camera1	0.451	0.462	0.457	0.619	0.358	0.454	0.473	0.449	0.461	0.656	0.367	**0.471**
Digital camera2	0.546	0.463	**0.501**	0.419	0.569	0.483	0.567	0.438	0.494	0.540	0.460	0.497
DVD player	0.334	0.685	0.449	0.430	0.452	0.441	0.461	0.487	0.474	0.468	0.507	**0.486**
Mp3 player	0.370	0.630	0.466	0.345	0.478	0.401	0.531	0.470	0.509	0.588	0.457	**0.515**

Table 3: Detailed performance comparison of USE DAN model with baseline GFLM-S and GFLM-W.



(a)

(b)

Figure 2: F_1 score plot for different methods, for (a) Digital camera1, (b) Medical datasets, over threshold between 0 and 1.

note that this higher recall has little practical value as the corresponding precision is low. On the other hand, GFLM-W and GFLM-S achieved comparatively high precision while preserving reasonable recall. For GFLM-W, GFLM-S, and *USE* the inference threshold (θ) was varied between 0 and 1 and then the maximum score is reported in the table. We have also presented performance of GFLM-W, GFLM-S, and *USE* for a fixed number of labels over different threshold in figure 2. At the end, results were *surprising* as *USE* was outperformed by simple topic-based inference techniques for zero-shot classification tasks, which motivated us to dig deeper into the reasons of *USE*'s score.

5.1 Why is *USE* Failing?

We performed a deeper investigation on whether *USE* can distinguish two closely related labels with a high semantic overlap, which inspired us to look at correlation heat-maps among different labels for each dataset. The correlation of two labels can be trivially computed using cosine similarity between two label embeddings (We would like to mention here that embeddings produced by the USE are approximately normalized). Figure 3 shows an example correlation heat-map of Digital camera1 dataset labels, where, darker color represent high correlation compared to the lighter one. For instance, embedding vector for ***Lens*** and ***Focus*** possess a higher correlation. Likewise, ***Size*** and ***Weight*** have high correlation as they are semantically close. In fact, we observed similar highly correlated labels for other datasets too. Due to space limitation, heatmaps of other datasets are presented in appendix A.3.

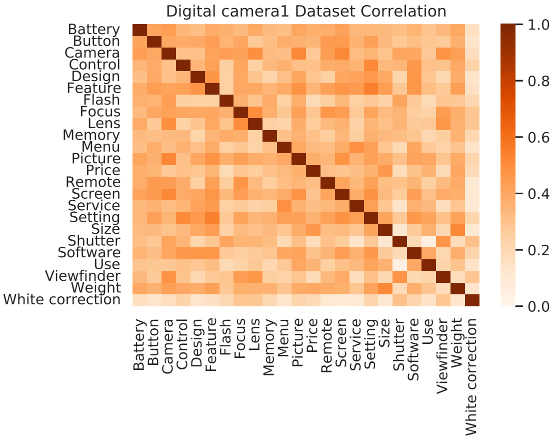


Figure 3: Correlation analysis of labels used in Digital camera1 dataset

Given these overlapping labels in our datasets, we hypothesised that *USE* is demonstrating suboptimal performance because it is failing to accurately distinguish between two labels with high semantic overlap. To test whether this is indeed the case, we greedily started reducing the number of labels. The motivation here is to analyze whether *USE* performance rises with decreasing number of overlapping labels and vice-versa.

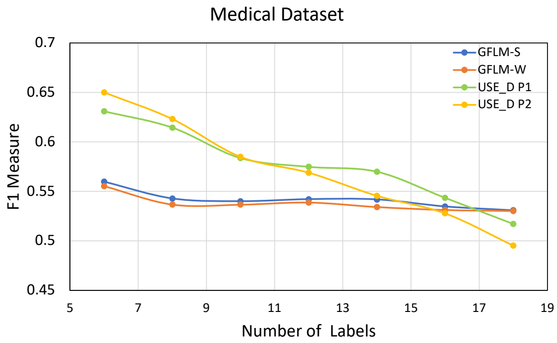


Figure 4: F_1 score plot for Medical dataset for descending number of labels

For removing the labels, we took a greedy approach where we first identified the highly correlated labels. At each iteration, we reduced 2-3 labels based on the semantic overlap and performed classification using the same method described in algorithm 1. The label count-performance tradeoff is better demonstrated via figure 4 for "Medical" dataset, (for rest of the datasets, results are presented in the appendix). It is evident from the trend of the performance that as we reduce the number of labels, performance clearly rises. Upon error analysis, we observed that for Medical dataset if an article is related to ***"Arthritis"*** and "***Pain Management"*** Universal Sentence Encoder labeled the article with ***"Osteoporosis", "Arthritis"*** and ***"Pain Management"***. The reason being ***"Arthritis"*** and ***"Osteoporosis"*** has high correlation / semantic similarity measure around 0.682. Reducing the label count moderated these kinds of scenarios. To be precise, when label ***"Osteoporosis"*** was excluded from the set, for the same article USE inferred ***"Arthritis"*** and ***"Pain Management"***. As a result, false positive counts minimise and performance uprise. We also continued the experiment with GFLM models with the reduced labels but we found that the performance was mostly stable in case of GFLM with little rise in F_1 score. This shows the GFLM models do not suffer for the high number/overlap of target labels.

6 DISCUSSION AND CONCLUSION

In this paper, we present a so-called "negative" result on *USE* in the context of "Zero-shot Text Classification". Our experimental results reveal that basic topic-based inference models outperformed *USE*-based inference, which is indeed surprising. Further investigation revealed that *USE* struggles to achieve good performance on zero-shot classification tasks with a large number of labels with high semantic overlap. On the other hand, simple topic based inference techniques were found to be pretty robust as a zero-shot classifier. One possible explanation for such performance may be attributed to the fact that topic-distribution vectors are constrained (sums to 1), while USE vectors are unbounded (real numbers). Such constrained representation of topic-vectors may make them superior in terms of their capability to distinguish between two highly overlapping labels compared to same for unbounded USE vectors, which were not trained following such constraints. In normal supervised learning settings, USE usually learn those distinctions from training labels, however, in case of zero-shot cases, that distinguishing capability is perhaps not developed well.

In summary, this paper highlights a limitation of the *USE* encoding technique and forms a cardinal basis for further research on the limitation of *USE*. Our findings also suggest that we may be still far away from a sentence encoding technique that is indeed "*universal*".

References

Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele. 2016. Multi-cue zero-shot learning with strong supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 59–68.

Meysam Asgari-Chenaghlu, Narjes Nikzad-Khasmakhi, and Shervin Minaee. 2020. Covid-transformer: Detecting covid-19 trending topics on twitter using universal sentence encoder. *arXiv preprint arXiv:2009.03947*.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Qi Chen, Wei Wang, Kaizhu Huang, and Frans Coenen. 2021. Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet of Things Journal*.

Ayesha Enayet and Gita Sukthankar. 2020. A transfer learning approach for dialogue act classification of github issue comments. *arXiv preprint arXiv:2011.04867*.

Jiaying Gong and Hoda Eldardiry. 2021. *Zero-Shot Relation Classification from Side Information*, page 576–585. Association for Computing Machinery, New York, NY, USA.

Tristan Hascoet, Yasuo Ariki, and Tetsuya Takiguchi. 2019. Semantic embeddings of generic objects for zero-shot learning. *EURASIP Journal on Image and Video Processing*, 2019(1):1–14.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2016. Generative feature language models for mining implicit features from customer reviews. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 929–938.

Hafsa Lattar, Aicha Ben Salem, and Henda Hajjami Ben Ghezala. 2020. Duplicate record detection approach based on sentence embeddings. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 269–274. IEEE.

Soumayan Bandhu Majumder and Dipankar Das. 2020. Detecting fake news spreaders on twitter using universal sentence encoder. In *CLEF*.

Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. 2017. Train once, test anywhere: Zero-shot learning for text classification. *arXiv preprint arXiv:1712.05972*.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Pablo Rivas and Marcus Zimmermann. 2019. Empirical study of sentence embeddings for english sentences quality assessment. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 331–336. IEEE.

Souvika Sarkar and Shubhra Kanti Karmaker. 2022. Concept annotation from users perspective: A new challenge. In *Companion Proceedings of the Web Conference 2022*, pages 1180–1188.

Sappadla Prateek Veeranna, Jinseok Nam, EL Mencía, and J Furnkranz. 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proceeding of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: Elsevier*, pages 423–428.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3090–3099. Association for Computational Linguistics.

Huang Xie and Tuomas Virtanen. 2021. Zero-shot audio classification via semantic embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1233–1242.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921. Association for Computational Linguistics.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1031–1040. Association for Computational Linguistics.

7 Ethics Statement

In this paper, we have discussed about behavior of Universal Sentence Encoder for Zero-shot Text Classification. Through this, we hope to assist new research direction. To the fulfilment of this goal, we have worked with seven different real-world datasets. We did not obtain any explicit approval as our intended contents were already published for educational/research purpose. We have not tried to identify any private information from the data in any way which can result in a privacy violation. Additionally, the data we used (publicly release) does not contain personal information (e.g., usernames of users). In the whole experiment, we only used open source packages and libraries, along with proper citations as required also in accordance with its acceptable use policy, and no additional permission was required.

A Appendix

A.1 Challenges of Zero-shot TC

A closer look into into the datasets revealed that they are comprised of articles with varying length and each article is a complex representation of various concepts, entities and events and most of the labels are not explicitly mentioned in the article and are thus “implicit” labels. The difference between the two can be further clarified through an example. We consider a label as explicit if the label name/phrase is explicitly mentioned in the article text. For example, the following sentence is from an article related to label ***Corona virus***, ***“Americans should feel much better about the corona virus coming under control”***, which mentions the label ***Corona virus*** explicitly in the text body. Whereas, for implicit cases, the label name is not directly mentioned in the article text, rather the label is somewhat implied. For example, the following sentence is taken from an article annotated with the label ***Women’s Health***, ***"Studies question: ban on alcohol during pregnancy."*** Here, the text does not contain the phrase ***Women’s Health***, yet a human can easily relate it to the same label. Recognizing implicit label is an arduous job. Probing our datasets, we ascertained significant portions of the data contains these implicit label, hence their accurate identification, is indeed very challenging, specially for "Zero-shot Text Classification" without any supervision.

To mitigate the issue of the ubiquity of implicit labels, we started to find alternative approaches. On further assessment, we realized that in cases where label names are not directly mentioned in the text, some informative keywords related to the label are always present in the article text. Indeed, each label can be imagined as a cloud of its informative keywords and different labels will essentially yield different word clouds. More interestingly, these informative keywords (word cloud) can be provided by the end user conducting the classification task. In fact, we realized this is what mostly happens in real-world cases. However, we did not have any end user involved in the task and also the keywords related to the labels were not readily available. Hence, we used TF-IDF heuristics and then extracted set of keywords for each label. For example, the articles related to label ***‘Women’s Health’*** yielded informative keywords like ***‘Pregnancy’, ‘Breast’, ‘Uterus’,‘Postpartum’, ‘Pregnant’, ‘Miscarriage’*** etc. This informative keywords are an important factor for the task and hence necessary.

A.2 Label Embedding Approaches

We have used 2 different approaches for computing label embedding. The consecutive sections discuss about different procedures for generating label embedding.

A.2.1 Label embedding using explicit annotated text (P1)

1. As discussed in algorithm 1, inputs are fed to pre-trained *USE*, such as article text and the labels with associated keywords.
2. Based on the labels and keywords "Explicit Annotator" module annotate some of the article which we consider as explicit annotated text. For an example, "The camera is great!!!", this review contains the the label "camera" explicitly, therefore "Explicit Annotator" marks the text as to be potentially connected to "camera".
3. These "Explicit Annotated Text" along with labels (in which user is interested) and candidate text (to be classified) are fed to Universal Sentence Encoder. Two separate vectors are generated by *USE*: a) *Text Embedding*: embedding generated for the candidate text, directly using *USE*; and b) *Label Embedding*: Label embedding is obtained by computing the average of all explicit annotated text. For an example, if the "Explicit Annotator" method identify 10 reviews based on labels and keywords search, which might be related to label "Camera" then we obtain 10 sentence embeddings and average them to get the label embedding for label "Camera".
4. Once the text and label embeddings have been computed, then semantic similarity between the text embedding and each label embedding is computed in terms of the cosine similarity.
5. Finally, based on a threshold technique, most relevant labels are inferred as the output.

A.2.2 Label embedding using label name and keywords (P2)

1. The input is same as stated in the A.2.1, article text and the label with associated keywords.
2. Also similar to previous method, two separate vectors are generated by *USE*: a) *Text Embedding*: sentence embedding generated on the candidate text, directly from *USE*; and b) *Label Embedding*: However, here label embedding is obtained by computing the average vector of label name embedding and keywords embed-

ding. For an example, if the label was "Sound" and set of associated keywords were "Audio", "Headphone", "Earbud" and "Earphone", then we compute the label embedding by taking average of label name ("Sound") and all the associated keywords ("Audio", "Headphone", "Earbud" and "Earphone") embeddings.

3. The procedure for final text classification is same as discussed in step 4 and 5 previously.

A.3 Correlation Analysis

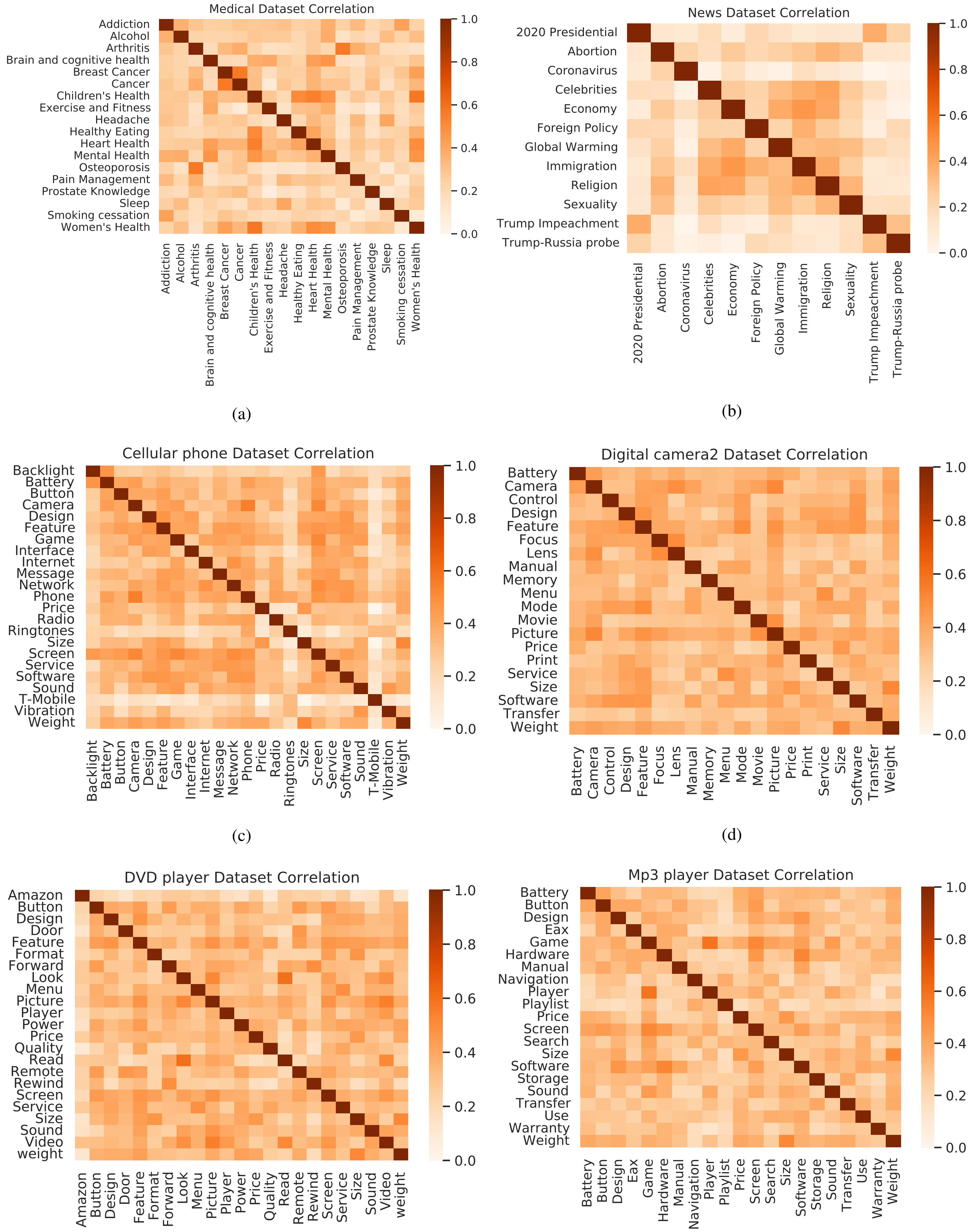
Heat maps for all datasets for correlation analysis has been presented in figure 5.

A.4 Performance comparison of GFLM and USE

Figure 6 present detailed comparison over all the methods for threshold between 0 to 1.

A.5 Label Vs Performance

Table 4 contains details for all datasets over different count of labels. Figure 7 is presented for showing label count vs performance trade-off.


Medical Dataset Correlation

(a)

News Dataset Correlation

(b)

Cellular phone Dataset Correlation

(c)

Digital camera2 Dataset Correlation

(d)

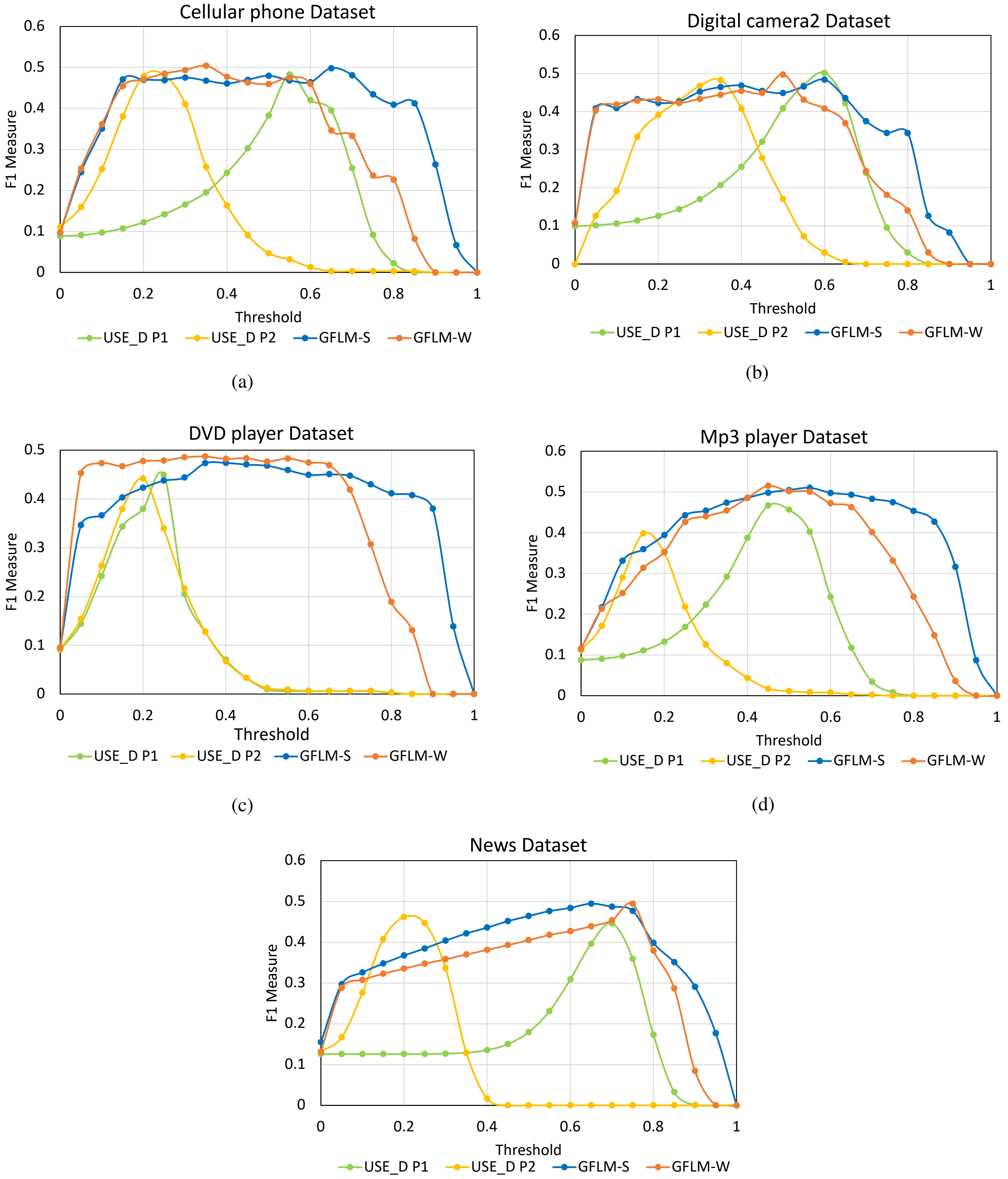
DVD player Dataset Correlation

(e)

Mp3 player Dataset Correlation

(f)

Figure 5: Correlation or semantic similarity heat-maps for (a) Medical, (b) News, (c) Cellular phone, (d) Digital camera2, (e) DVD player and (f) Mp3 player datasets.



(a)

(b)

(c)

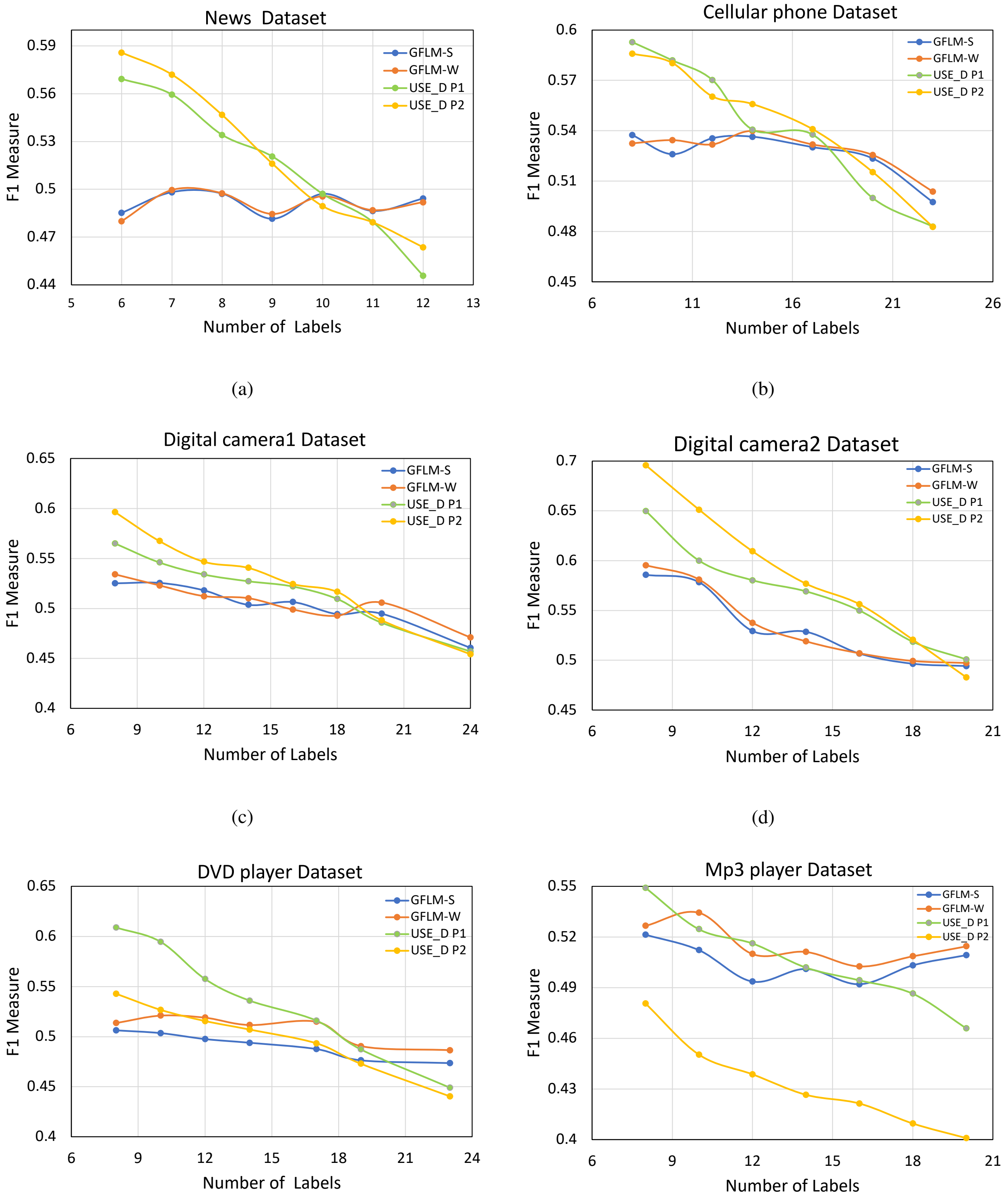
(d)

(e)

Figure 6: F_1 score plot for different methods for (a) Cellular phone, (b) Digital camera2 , (c) DVD player, (d) Mp3 player, (e) News datasets, over threshold between 0 and 1.

Dataset	**Label Count**	**GFLM-S**	**GFLM-W**	$\mathbf{USE_{D}^{P1}}$	$\mathbf{USE_{D}^{P2}}$
Medical	18	0.531	0.530	0.517	0.495
	16	0.888	0.531	0.544	0.527
	14	0.542	0.534	0.569	0.546
	12	0.542	0.539	0.574	0.569
	10	0.540	0.537	0.584	0.584
	8	0.543	0.537	0.615	0.623
	6	0.559	0.556	0.631	0.650
News	12	0.494	0.491	0.445	0.464
	11	0.486	0.487	0.479	0.479
	10	0.497	0.495	0.498	0.489
	9	0.482	0.485	0.521	0.516
	8	0.497	0.497	0.534	0.547
	7	0.498	0.496	0.559	0.572
	6	0.485	0.480	0.569	0.585
Cellular phone	23	0.498	0.504	0.483	0.482
	20	0.524	0.526	0.500	0.515
	17	0.530	0.532	0.538	0.541
	14	0.536	0.540	0.541	0.556
	12	0.536	0.532	0.570	0.560
	10	0.526	0.534	0.582	0.580
	8	0.537	0.533	0.592	0.586
Digital camera1	24	0.461	0.471	0.457	0.454
	20	0.495	0.506	0.486	0.488
	18	0.494	0.493	0.509	0.517
	16	0.506	0.499	0.522	0.524
	14	0.504	0.510	0.527	0.541
	12	0.518	0.512	0.534	0.547
	10	0.526	0.523	0.546	0.567
	8	0.525	0.534	0.565	0.596
Digital camera2	20	0.494	0.497	0.501	0.483
	18	0.497	0.499	0.519	0.521
	16	0.507	0.507	0.550	0.556
	14	0.529	0.519	0.569	0.577
	12	0.529	0.538	0.580	0.609
	10	0.578	0.581	0.600	0.651
	8	0.586	0.596	0.650	0.696
DVD player	23	0.474	0.486	0.449	0.440
	19	0.476	0.491	0.487	0.473
	17	0.488	0.515	0.516	0.493
	14	0.494	0.512	0.536	0.507
	12	0.497	0.519	0.557	0.516
	10	0.503	0.521	0.594	0.527
	8	0.506	0.514	0.609	0.543
Mp3 player	21	0.509	0.515	0.466	0.401
	18	0.503	0.509	0.487	0.410
	16	0.492	0.503	0.494	0.421
	14	0.501	0.511	0.502	0.427
	12	0.494	0.510	0.516	0.439
	10	0.512	0.534	0.525	0.450
	8	0.521	0.527	0.549	0.481

Table 4: Performance comparison of all the datasets over varying number of labels. Results presented in the table is for the DAN architecture over 2 different embedding process P1 and P2.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: F_1 score plot for (a) News, (b) Cellular phone, (c) Digital camera1, (d) Digital camera2, (e) DVD player, and (f) Mp3 player over different label count.