What Do Users Care About?
Detecting Actionable Insights from User Feedback

Kasturi Bhattacharjee[1], **Rashmi Gangadharaiah**[1], **Kathleen McKeown**[1,2], **Dan Roth**[1,3]

[1]AWS AI Labs
[2]Columbia University
[3]University of Pennsylvania
{kastb,rgangad,mckeownk,drot}@amazon.com

Abstract

Users often leave feedback on a myriad of aspects of a product which, if leveraged successfully, can help yield useful insights that can lead to further improvements down the line. Detecting actionable insights can be challenging owing to large amounts of data as well as the absence of labels in real-world scenarios. In this work, we present an aggregation and graph-based ranking strategy for unsupervised detection of these insights from real-world, noisy, user-generated feedback. Our proposed approach significantly outperforms strong baselines on two real-world user feedback datasets and one academic dataset.

1 Introduction

Collecting vast amounts of user feedback on products and services is a common practice these days for a multitude of companies. This can prove to be a rich resource for product owners to improve the quality of their product offerings, correct failures and gauge the general performance of their product from both implicit and explicit signals that might be present in the feedback. However, in most cases, including the one we address in this work, the feedback is unstructured and voluminous, and can therefore remain underutilized for the most part. In our particular use-case, we receive thousands of textual feedback daily on average [1], and it is time-consuming and laborious for product owners to manually extract actionable insights from them.

Users leave feedback on a variety of aspects they experience in the course of using the product (Table 1). These consist of functionalities they find useful (*calculate total size*), issues they encounter when attempting to perform an action (*scroll sideways*), requests around enabling certain features (*sort by Date Modified*), and so on. In this work, our goal is to capture short informative phrases, or ***themes*** which reflect *actionable* insights in the feedback. In capturing these actionable insights, we wish to focus on desired actions that users want to be able to carry out (e.g. *scroll* sideways).

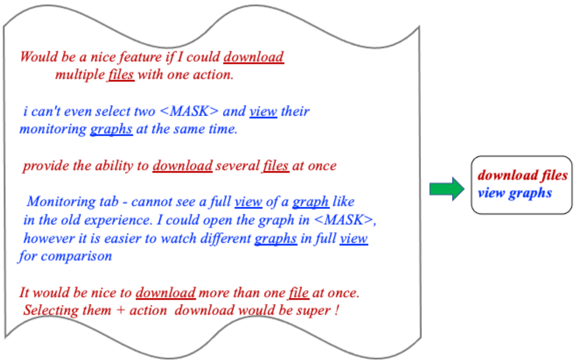
User Feedback Examples
loved the new calculate total size!
Please let me sort by Date Modified
Why can't I scroll sideways on my mac specifically in the new console?

Table 1: Feedback examples from real-world (internal) datasets that illustrate user issues and feature requests when performing an action (underlined).

Owing to the influx of this data in large amounts, and the cost of annotations, it often remains *unlabeled*. Therefore, in this work, we present an *unsupervised* framework to detect actionable insights from such data. In order to capture these insights from an aggregated view of the data, we propose the following two-step approach: a) *aggregating* similar feedback such that each cluster represents coherent insights; b) *detecting themes* from clusters that are pertinent to actionable insights. For instance, in Figure 1, the red cluster consists of feedback expressing users' need to be able to download several files at one time, for which a *possible* theme could be *download files*. As seen in these examples, the desired themes may consist of non-contiguous tokens appearing in text and typically contain a verb mentioning the action. Prior work (see Section 2) uses keyphrase extraction which extracts contiguous words within a noun phrase and thus, cannot capture the kind of actionable insights we find in our data. We utilize unsupervised clustering algorithms for the aggregation step, and a graph-based ranking strategy for the theme detection step.

[1] We take our responsibility to protect customer content extremely seriously. For this research, we followed Amazon's Customer Content policy guidelines.

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 239 - 246
July 10-15, 2022 ©2022 Association for Computational Linguistics



Would be a nice feature if I could download multiple files with one action.

i can't even select two <MASK> and view their monitoring graphs at the same time.

provide the ability to download several files at once

Monitoring tab - cannot see a full view of a graph like in the old experience. I could open the graph in <MASK>, however it is easier to watch different graphs in full view for comparison

It would be nice to download more than one file at once. Selecting them + action download would be super !

Figure 1: Our pipeline is illustrated above. User feedback documents are first grouped into various clusters. Here, two clusters (blue and red), each representing different insights are shown. For each cluster, we automatically identify a "theme" (right part of the figure) that concisely captures the desired actions expressed in the cluster, e.g. *download files* for the red cluster; *view graphs* for the blue cluster. <*MASK*> tokens are inserted to maintain anonymity when displaying the examples above.

Contributions of the paper:

- We propose a novel approach of identifying actionable user insights in an unsupervised manner. We do so by framing the problem as a clustering and cluster theme detection problem.
- Our approach is unique in the utilization of graph-based ranking in the identification of cluster themes, especially focused on capturing actions that users want to perform.
- We find our method to significantly outperform baselines on two real-world datasets and one academic dataset.

2 Related Work

There is limited work that focuses on uncovering insights from *real-world* user feedback data. Most of the work involves labeled academic datasets, requiring approaches that involve some form of supervision. For instance, Lin et al. (2012) involves a weakly supervised joint sentiment-topic model that detects sentiment and topic simultaneously from text, applied to two labeled academic datasets. Approaches that are unsupervised (Qiu et al., 2021; Liu et al., 2010) are largely based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which yields distribution of unigrams as topics. Although Qiu et al. (2021) adopt an unsupervised strategy, it has not been explored on user feedback data. These approaches are not directly applicable to our problem setting since we require short phrases focusing on the performance of an *action* by the user.

Approaches such as TextRank (Kazemi et al., 2020), SingleRank (Wan and Xiao, 2008), ExpandRank (Wan and Xiao, 2008), TopicRank (Bougouin et al., 2013), TopicalPageRank (Sterckx et al., 2015), PositionRank (Florescu and Caragea, 2017), Bi-LSTM-CRF Sequence Labeling (Alzaidy et al., 2019), FACE (Chau et al., 2020), and MultipartiteRank (Boudin, 2018) have been applied to the task of key phrase extraction from documents as opposed to theme detection from clusters. For cluster-labeling, there has been work around using keyword extraction from clusters, utilizing WordNet synsets to expand the keywords, followed by a selection procedure to assign the final label (Poostchi and Piccardi, 2018; Chang and McKeown, 2019). None of these approaches focus on the extraction of short *actionable* phrases, a.k.a. *themes*, as is our use case. In most of these approaches, candidate key phrases are assumed to appear in contiguous positions in a document and are concatenated to form phrases. As described earlier (Figure 1), it is unrealistic to make such assumptions on real-world user feedback data and our proposed approach considers non-contiguous candidate phrases as well.

3 Data

Split	# of samples	# of samples per intent
Train	15K	100
Dev	3K	20
Test	4.5K	30

Table 2: CLINC150 Data Statistics

In this work, we use two internal unlabeled datasets containing user feedback in English on two product offerings. The feedback collection pipeline has opt-in and opt-out mechanisms in place to allow the user to decide whether their data can be used for further analysis. User-specific information has been removed from these datasets for privacy and confidentiality reasons. These datasets vary in content based on the specific product they contain feedback about, and in the number of feedback documents contained in each. We refer to them as $\mathbf{Prod_1}$ and $\mathbf{Prod_2}$. $\mathbf{Prod_1}$ received orders of magnitude more feedback than $\mathbf{Prod_2}$. For exploration purposes we sampled about 10K documents for $\mathbf{Prod_1}$ and 1.5K documents for $\mathbf{Prod_2}$. In addition,

Intent Label	Document
find phone	*i need your help finding my lost phone*
book hotel	*i'm inquiring about the availability of a room that fits 10 people from monday to tuesday in manhattan*
schedule maintenance	*please find someone who specializes in cars, my check engine light has turned on*

Table 3: Examples with various intent labels from **CLINC150** Dataset. **Note that labels are utilized merely for evaluation purposes.**

we conduct evaluations and report results on an intent classification dataset - **CLINC150** (CC 3.0) (Larson et al., 2019; Zhang et al., 2020) that contains English utterances labeled with one of 150 intents, thereby containing *document-level* labels. The data contains utterances from 10 domains, e.g. Banking, Travel, Kitchen & Dining etc. Table 3 contains utterance examples. The intent labels in this data (e.g. *find phone, book hotel, schedule maintenance, etc.*) are similar in form to the cluster themes we aim to discover from our product feedback data, which makes it a good candidate for evaluating our approach. We obtain proxy ground-truth labels (details in Section 5.3) from the data to evaluate and report metrics on this dataset. **Note that the labels were used for the sole purpose of evaluation and not training, since the real-world use case is in an unsupervised setting.** We use the same train/dev/test splits provided with this dataset, statistics of which are reported in Table 2.

4 Methodology

In this section, we describe our proposed approach for extracting themes that help in discovering insights from user-generated text (outlined in Figure 2). In order to discover coherent emerging insights from the vast amounts of user-generated data, we first aggregate semantically similar feedback using clustering algorithms, and extract a *representative set* of documents per cluster. This is described in Section 4.1. Thereafter, themes for these clusters are generated as detailed in Section 4.2. The entire pipeline is unsupervised.

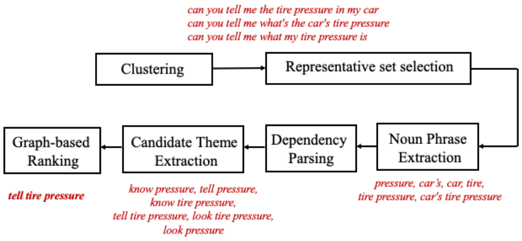

can you tell me the tire pressure in my car
can you tell me what's the car's tire pressure
can you tell me what my tire pressure is
Clustering
Representative set selection
Noun Phrase Extraction
pressure, car's, car, tire, tire pressure, car's tire pressure
Dependency Parsing
Candidate Theme Extraction
know pressure, tell pressure, know tire pressure, tell tire pressure, look tire pressure, look pressure
Graph-based Ranking
tell tire pressure

Figure 2: Outline of proposed approach.

4.1 Document Aggregation & Representative Set Selection

Documents are embedded using Sentence-BERT (SBERT) (Reimers and Gurevych, 2019), which we find to capture semantic similarity well even for our internal datasets. k-means (MacQueen et al., 1967) is selected as our clustering approach of choice. Additionally, we explore the use of DNN-based clustering approaches such as Deep Embedded Clustering (**DEC**) (Xie et al., 2016) which has been shown to outperform k-means for a few academic text classification datasets (e.g. REUTERS (Lewis et al., 2004)). When applied to our real-world user-generated text, we find DEC to perform well on the larger dataset, Prod$_1$ but not on the smaller dataset Prod$_2$ (Table 4), while k-means performs well across both datasets. Thus, for the remainder of the paper, we report results using k-means as the clustering strategy.

Based on the hypothesis that the centroid of a cluster is representative of the overall cluster itself, we rank documents based on their proximity to the centroid for each cluster. 10 documents closest to the centroid per cluster are subsequently considered for cluster label detection. We chose 10 documents as it provided a good balance between having good representative members of the cluster while also ensuring that we have very few noisy cluster members, if any. Henceforth, we refer to these as the **representative set** of a cluster.

4.2 Unsupervised Cluster Theme Identification

Here, we describe the procedure for cluster theme identification. As previously mentioned, our themes could consist of non-contiguous tokens. We begin by extracting candidate themes using a dependency parsing approach, followed by a graph-based ranking strategy to assign a theme per cluster.

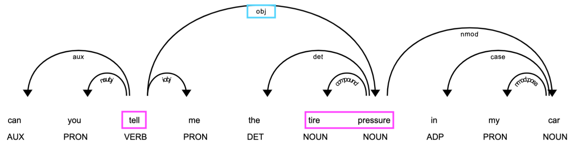


Figure 3: Example of dependency parsing output on CLINC150 utterance.

4.2.1 Cluster Theme Candidate Extraction

- **Noun Phrase Extraction:** Nouns, Proper Nouns and Noun Phrases are extracted from the representative set per cluster. All pronouns that occur in the beginning/end of noun phrases are removed, e.g. *my check engine light* converted to *check engine light*. This is done to capture more generic noun phrases. Those that occur > 2 times throughout the ranked set are retained.
- **Dependency Parsing:** We run a Dependency Parser and extract all verbs for which any of the above selected set of nouns and noun phrases are a *nominal subject* or *object*. For instance in Fig. 3, *tire pressure* is an object for verb *tell*.
- **Candidate Theme Extraction:** Phrases of the form <VERB, NOUN> are thus constructed. For cases where the Noun is part of a previously selected noun phrase, we expand the phrases to <VERB, NP>. These act as *candidate themes* for a given cluster.

4.2.2 Graph-based Ranking for Theme Identification

Graph-based ranking algorithms are often employed in approaches for unsupervised document summarization to measure the importance of a sentence for inclusion in a summary (Erkan and Radev, 2004; Zheng and Lapata, 2019). We apply a similar approach for cluster theme detection. Inspired by these approaches, we construct a graph per cluster, where the nodes consist of the candidate themes and the edge weights capture the semantic similarity between pairs of themes, obtained using cosine similarity between SBERT embeddings of corresponding themes. We then use a graph-based ranking strategy, PageRank (Brin and Page, 1998) and assign the phrase with the highest rank as the cluster theme.

5 Experiments

In this Section, we describe the experimental details of the pipeline, and provide details on the baselines we compare with.

5.1 Data Processing

Since user-generated text tends to be noisy in nature, we preprocess our internal datasets beforehand. This includes converting to lowercase, removing URLs, special characters, and removing text consisting only of digits. For CLINC150, the only pre-processing performed is to replace underscores in the intent labels with spaces, i.e. *book_hotel* converted to *book hotel*, since the generated themes are of a similar form.

5.2 Baselines

We use two baselines to compare with, which are described below.

Poostchi and Piccardi (P&P) This work proposes an approach for cluster labeling by leveraging word embeddings and the synonymy and hypernymy relations in the WordNet (Miller, 1995) lexical ontology. Similar to Chang and McKeown who adapt this method for their clusters, we perform the following steps for each of our clusters. We extract keywords using RAKE (Rose et al., 2010) from the representative set per cluster, to ensure a fair comparison with our methodology. Hypernyms of the component words (restricted to Nouns) of these keywords are obtained, expanded by synonyms (via *synsets*, WordNet's synonym sets). We use *CentHyp* - the best strategy as per Poostchi and Piccardi, to assign the final cluster label. This selects hypernyms that are most central w.r.t. the centroid of the cluster. SBERT embeddings are used for this purpose, to ensure a fair comparison with our approach. Since we could not find offical code for this work, we used our own implementation.

Random baseline We also compare with a random baseline in which the cluster theme is selected at random from the generated set of cluster themes using our proposed approach (Section 4.2.1).

Dataset	**Mean Accuracy Score (Human Eval)**	
	k-means	**DEC**
Prod$_1$	90.0	80.0
Prod$_2$	65.0	45.0

Table 4: k-means and DEC clustering algorithms compared on the internal datasets. Annotators provide a score of 0 or 1 to the clusters, based on whether they agree with the quality. Average accuracy per annotator is computed. Scores reported in this table are an average of the annotator scores.

5.3 Evaluation Strategies & Metrics Reported

CLINC150 For this dataset, we leverage the document-level intent labels to generate a *proxy* groundtruth label per cluster. The same representative set (as in Section 4.1) is selected per cluster and the cluster theme is determined by a majority vote over the intent labels of these documents. For the rare case where there is no clear majority, we consider the label of the centroid to be the cluster label. We compute METEOR (Denkowski and Lavie, 2014) and BERTScore (Zhang et al., 2019) metrics w.r.t. the model outputs and proxy labels for our baselines as well as the proposed approach.

Data	Method	METEOR	BERTScore
Dev	P&P	21.30 ±0.69	0.8892 ±0.0033
	Random	21.12 ±0.27	0.8921 ±0.0021
	Ours	**25.79** ±0.55	**0.8962** ±0.0023
Test	P&P	20.94 ±1.37	0.8912 ±0.0034
	Random	19.59 ±1.47	0.8955 ±0.0023
	Ours	**25.31** ±0.69	**0.9026** ±0.0018

Table 5: Mean and standard deviation for METEOR and BERTScores reported for the proposed approach (Ours) against baselines, P&P (Poostchi and Piccardi) and Random on CLINC150, for 3 runs.

Internal Datasets Since Prod_1 and Prod_2 do not contain labels either on the document or cluster level, we utilize human annotations to evaluate the efficacy of our methodology w.r.t. the baselines. We employ 2 internal annotators who are presented with the cluster themes generated on both datasets, and the representative set of documents per cluster that the themes were detected from. An annotator votes 1 if they completely agree with the cluster theme, 0 if they completely disagree. The final scores are an average of the scores of both annotators. IAA (Cohen's kappa) score is 0.72.

5.4 Modeling Details

We use Stanford Stanza (Qi et al., 2020) for POS tagging and dependency parsing, and scikit-learn's (Pedregosa et al., 2011) k-means implementation for clustering. Using input number of clusters to be the same as the number of intent labels $k = 150$ yields the best results for CLINC150. For our internal datasets, $k = 150$ for Prod_1 and $k = 25$ for Prod_2 are used. For PageRank, we use networkx's (Hagberg et al., 2008) package, with maximum number of iterations set to 100. CPU-based computing instances are used for both baselines and our methodology.

Dataset	Method	Human Evaluation Score
Prod_1	P&P	32.89
	Random	29.70
	Ours	**72.60**
Prod_2	P&P	16.67
	Random	8.00
	Ours	**40.00**

Table 6: Performance of the proposed approach (Ours) against baselines, P&P (Poostchi and Piccardi) and Random on our internal datasets.

(Proxy) Cluster Theme	Baseline (P&P) Prediction	Our Prediction
pay bill	electric bill	**pay bill**
calendar	check	**check calendar**
meeting schedule	today	**schedule meeting**
insurance change	insurance policy	**update insurance policy**
shopping list	**shopping list**	buy milk
change accent	**change**	change voice

Table 7: Comparing cluster theme predictions from P&P (Poostchi and Piccardi) & our approach on CLINC150 test set. Themes in bold are those that are qualitatively most similar to the proxy cluster theme.

6 Results

Model performance on CLINC150 Table 5 illustrates the performance of the baseline models and our proposed approach on the dev and test splits of the CLINC150 dataset, over 3 experimental runs.

Dev set: Statistical significance tests conducted show that on the dev set, our methodology significantly outperforms random baseline on METEOR

Representative set of documents per cluster	P&P	Ours
- The new monitoring tab is very poor. It doesn't show enough data and you can't click on graphs to get detailed views. *- I cannot see the monitoring of 2 <MASK> <MASK> side by side. Unreliable reporting of metrics, graphs are sometime unavailable.* *- i can't even select two <MASK> and view their monitoring graphs at the same time.*	detailed view	**view graphs**
- Need to see the <MASK> alarms setup per <MASK> like before instead of loosing this functionality in a new <MASK>. *- I miss the alarm status ""button"" that shows all alarms connected to <MASK> <MASK>* *- On the new <MASK> dashboard all alarms connected to <MASK> <MASK> do not show as it does in the old console. On the old console you get a direct view if any of your <MASK> has any issues while we on the new get that there are no alarms for the <MASK>.*	alarms	**shows alarms**

Table 8: Comparing cluster themes from the baseline P&P method (Poostchi and Piccardi, 2018) vs ours. The documents are in ascending order of proximity to centroid. Text in bold highlights themes that best capture the action being performed by a user.

score by 4.67 points on average (p-value 0.0004). On mean BERTScore, we outperform the random baseline by 0.0041 points (p-value 0.1449). Compared with (Poostchi and Piccardi, 2018), we find our methodology to yield a significantly better performance on METEOR score - an increase of 4.49 points (p-value 0.003). Further, we outperform Poostchi and Piccardi (2018) on BERTScore by 0.007 points (p-value 0.148).

Test set: We significantly outperform both baselines - random and Poostchi and Piccardi (2018), by 5.72 points (p-value 0.0076) and 4.37 points (p-value 0.0159) respectively, on METEOR score. Performance gains obtained using our method over both baselines for BERTScore are also statistically significant - an increase of 0.0071 points (p-value 0.0293) w.r.t. random baseline and that of 0.0114 points (p-value 0.0132) w.r.t. Poostchi and Piccardi (2018).

Model performance on Internal datasets Table 6 demonstrates the significant boost in performance our proposed approach provides over the baselines, as measure by human evaluation scores. We find our approach to outperform both baselines by a large margin for both datasets. On Prod_1, we improve upon the random baseline by 42.9 points and upon Poostchi and Piccardi (2018) by 39.71 points. For Prod_2, we obtain an improvement of 32 points w.r.t. the random baseline and 23.33 points as compared to Poostchi and Piccardi (2018).

Error Analysis In Tables 7 and 8, we present examples comparing the cluster theme predictions from Poostchi and Piccardi (2018) with ours on the CLINC150 test set, and our internal datasets, respectively. Our method is able to yield more descriptive phrases as cluster themes, that help capture the action being performed. In comparison, the baseline captures shorter and less descriptive phrases. For instance, our method generates themes such as ***pay bill*** and ***update insurance policy*** on clusters from the CLINC150 test set, where the corresponding baseline themes are *electric bill* and *insurance policy*, respectively. Similarly, for a cluster from the internal dataset (Prod_1), the theme assigned by our proposed approach is ***show alarms***, whereas the theme detected by the baseline method is *alarms*.

7 Conclusion & Future Work

This work addresses the problem of discovering actionable insights from unlabeled real-world user feedback data, in an unsupervised fashion. Data is clustered into groups containing coherent insights, followed by theme detection per cluster using a graph-based ranking approach. Experiments conducted on two real-world user feedback datasets as well as an academic dataset show our proposed approach to significantly outperform baselines by a large margin. In the future, we would expand the scope of our work to datasets with other characteristics and distributions (e.g. review datasets) to study the applicability of our approach to those use-cases. Further, we would explore the use of generative models to obtain abstractive themes from data.

References

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*, WWW '19, page 2551–2557, New York, NY, USA. Association for Computing Machinery.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 667–672, New Orleans, Louisiana. Association for Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Serina Chang and Kathleen McKeown. 2019. Automatically inferring gender associations from language. *arXiv preprint arXiv:1909.00091*.

Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. 2020. Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*, pages 1 – 27.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.

Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Ashkan Kazemi, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Biased textrank: Unsupervised graph-based content extraction.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Chenghua Lin, Yulan He, Richard Everson, and Stefan Ruger. 2012. Weakly supervised joint sentiment-topic detection from text. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1134–1145.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 366–376, Cambridge, MA. Association for Computational Linguistics.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

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. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Hanieh Poostchi and Massimo Piccardi. 2018. Cluster labeling by word embeddings and WordNet's hypernymy. In *Proceedings of the Australasian Language Technology Association Workshop 2018*, pages 66–70, Dunedin, New Zealand.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

John Xi Qiu, Adam Faulkner, and Aysu Ezen Can. 2021. Towards theme detection in personal finance questions.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 121–122, New York, NY, USA. Association for Computing Machinery.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 855–860. AAAI Press.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Jian-Guo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. *arXiv preprint arXiv:2010.13009*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. *arXiv preprint arXiv:1906.03508*.