Evaluating Unsupervised Text Embeddings on Software User Feedback

Peter Devine, Yun Sing Koh, Kelly Blincoe

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

Overview in one sentence

Comparing various forms of text embedding to find which are best at grouping similar user feedback together

User feedback: What is it?

[redacted]

★★★★☆ August 31, 2021

125

Awesome game...been playing since galaxy s3...i love how they changed it with time...my only problem is the online connection needed to store your progress...sometimes it's a pain and you end up losing your current p

[redacted] 1h

aye yo @Snapchat we rly need a reply all button plz and thanks😘

VLC and MP4

by [redacted]

I downloaded the latest release of VLC but it does not read MP4 files. What did I do wrong? Is this a restriction?

Thanks

Issues with audio on app (self.netflix)

2

submitted 10 hours ago by [redacted]

Just today I have been having issues with the audio cutting out every minute or so but the video keeps playing. I updated the app, turned my phone off and on and closed all apps that were running in the background. It might be my internet connection but I thought I would ask if anyone else was having/ had this issue.

comment share save hide report

User feedback: Why is it important?

- Reaction of users to a piece of software

- From diverse sources:
  - app reviews (Pagano et al., Chen et al., Di Sorbo et al.)
  - tweets (Guzman et al., Nayebi et al., Williams et al.)
  - forum posts (Tizard et al.)
  - Reddit posts (Ali Khan et al.)

- Contains requirements relevant data like bug reports, feature requests useful to developers

Text embeddings

- Converts text into quantifiable, comparable numbers

```
“The app crashes when I turn it on”
                ↓
    [0.2, -0.3, 0.8, ...]
```

Text embeddings

We focus on 4 key families of text embeddings:

- Word-frequency methods (TF-IDF[1], BOW)
- Topic models (LDA[2], BTM[3], GSDMM[4])
- Averaged word embeddings (GloVe[5], ExtVec[6], USIF[7], other[8])
- Transformer based models (SBERT[9], USE[10], LaBSE[11])

Text embeddings - Word frequency

- Models used - Bag of words, TF-IDF
- Variations - remove stopwords, include bigrams

	The	app	crashes	often	I	use
“The app crashes often” →	1	1	1	1	0	0
“I use the app often” →	1	1	0	0	1	1

Text embeddings - Topic models

- Models used - LDA, BTM, GSDMM
- Creates {5, 13, 50} topics based on co-occurrence of words in a document within a corpus
- Characterises documents based on terms by distribution over topics

`“I would like to have a dark mode option”`

Topic 1	**Topic 2**	**Topic 3**
60%	35%	5%

Text embeddings - Avg. word embeddings

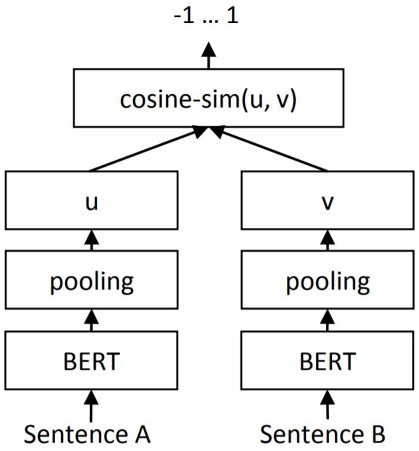
- Models used - GloVe, ExtVec, USIF, Levy and Goldberg
- Each word is given a pre-trained embedding, then averaged over the document

```
"The" -      [0.1, -0.3, 0.1, ...]
"App" -      [0.3, -0.5, 0.1, ...]
"Broke" -    [0.2, -0.1, 0.4, ...]

Embedding -  [0.2, -0.3, 0.2, ...]
```

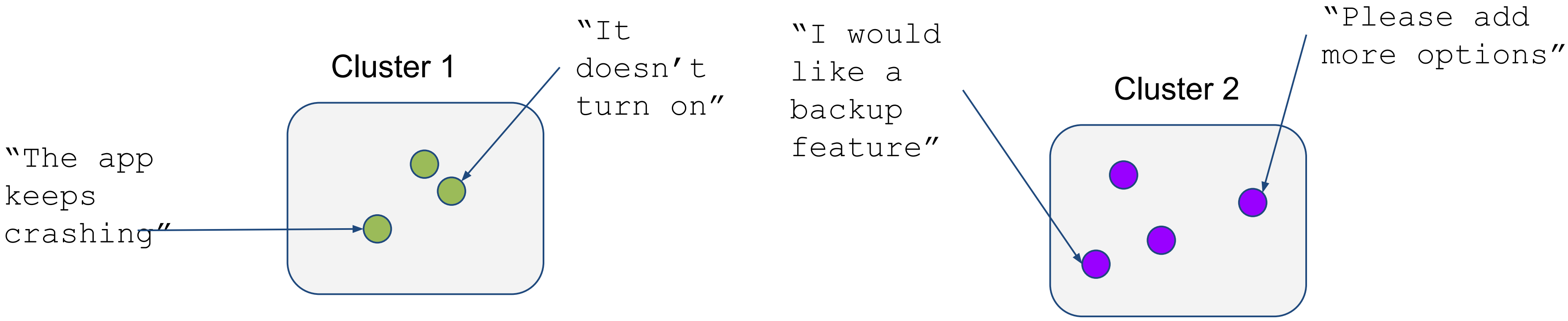
Text embeddings - Transformer models

- Models used - USE, SBERT, LaBSE
- Transformer based deep neural network pre-trained on language tasks (E.g. semantic similarity matching) generate an embedding for the whole piece of text



Text embeddings in user feedback

- Previous work (E.g. MERIT[1], CLAP[2]) has used text embeddings to cluster feedback into unsupervised groups
- Has used word frequency, topic modelling approaches



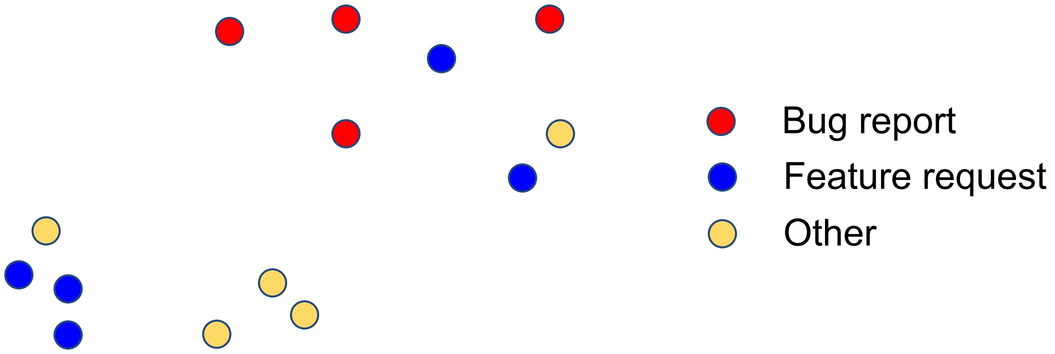
1. Gao et al. **Emerging App Issue Identification via Online Joint Sentiment-Topic Tracing**. TSE 2021 Apr 28.
2. Villarroel et al. **Release planning of mobile apps based on user reviews**. ICSE 2016 May 14 (pp. 14-24). IEEE.

Research gap

- No research done on evaluating which embedding method is best for unsupervised clustering of user feedback

Method

- Use 7 class labelled user feedback datasets (labelled A-G) to test the 4 text embedding classes
- We measure the distance of feedback with same class compared to feedback with different classes.



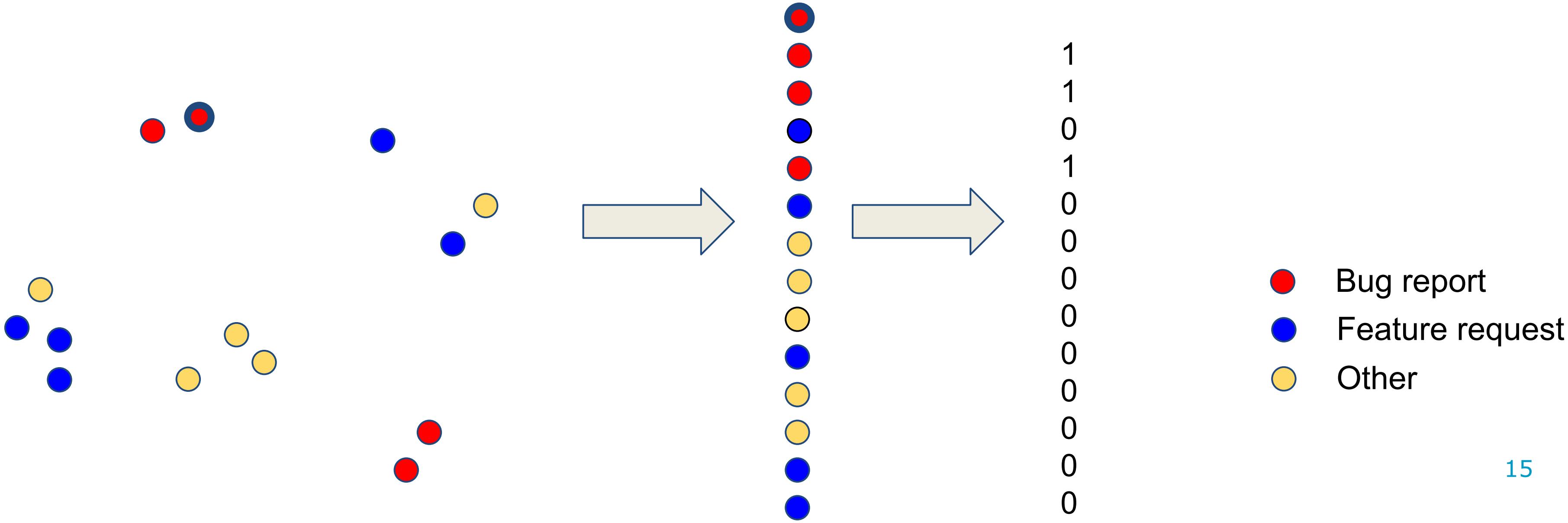
Method

TABLE I
DETAILS OF THE SEVEN CLASS-LABELLED USER FEEDBACK DATASETS USED FOR EVALUATING TEXT EMBEDDING METHODS

	Source	Feedback platform	Number of apps	Label set	Dataset size
A	Chen et al.[11]	Google Play Store reviews	4	Informative, Non-informative	11,340
B	Ciurumlelea et al.[13]	Google Play Store reviews	17	Resources, Usage, Compatibility, Pricing, Protection, Other	1,538
C	Guzman et al.[24]	Google Play Store reviews	7	Bug report, Noise, Usage scenario, Praise, Complaint, Feature shortcoming, Feature strength, User request	4,401
D	Maalej et al.[36]	Google Play Store reviews, Apple App Store reviews	24	Bug, Feature, User experience, Rating	488
E	Scalabrino et al.[44]	Google Play Store reviews	13	Feature, Performance, Usability, Security, Energy, Bug	702
F	Tizard et al.[47]	Forum posts	3 (2 apps, 3 forums)	User setup, Question on application, Requesting more information, Feature request, Non-informative, Malfunction confirmation, Question on background, Help seeking, Attempted solution, Application usage, Praise for application, Acknowledgement of problem resolution, Agreeing with the feature request, Agreeing with the problem, Limitation confirmation, Application guidance, Dispraise for application, Apparent bug, Other	3,654
G	Williams et al.[50]	Twitter posts	10	Feature, Bug, Other	3,907

Method - Evaluation

- Metrics - mean reciprocal rank (MRR) and mean normalised discounted cumulative gain (NDCG)



Results - MRR

	A	B	C	D	E	F	G
Best transformer model	**0.941**	**0.899**	**0.865**	**0.888**	**0.831**	**0.674**	**0.817**
Best avg. word embed model	0.905	0.887	0.816	0.838	0.773	0.582	0.727
Best word frequency model	0.902	0.879	0.783	0.825	0.788	0.601	0.758
Best topic model	0.863	0.840	0.749	0.817	0.743	0.497	0.686
Random baseline	0.730	0.753	0.555	0.774	0.636	0.354	0.590

Results - MRR (Transformers)

	A	B	C	D	E	F	G
USE	**0.941**	0.897	**0.865**	0.872	**0.831**	0.665	**0.817**
S-RoBERTa	0.919	0.892	0.859	**0.888**	0.787	0.642	0.782
S-BERT	0.910	0.889	0.847	0.885	0.781	0.636	0.770
LaBSE	0.920	**0.899**	0.828	0.853	0.772	**0.674**	0.764

Results - NDCG

	A	B	C	D	E	F	G
Best transformer model	**0.944**	**0.904**	**0.888**	**0.907**	**0.836**	**0.744**	**0.855**
Best avg. word embed model	0.928	0.893	0.844	0.856	0.794	0.706	0.815
Best word frequency model	0.922	0.894	0.831	0.844	0.800	0.698	0.821
Best topic model	0.933	0.877	0.834	0.843	0.783	0.678	0.811
Random baseline	0.903	0.838	0.768	0.821	0.738	0.641	0.784

Results - NDCG (Transformers)

	A	B	C	D	E	F	G
USE	0.941	**0.904**	0.871	0.886	**0.836**	**0.744**	**0.855**
S-RoBERTa	**0.944**	0.894	**0.888**	**0.907**	0.812	0.732	0.842
S-BERT	0.942	0.894	0.883	0.901	0.804	0.728	0.836
LaBSE	0.932	0.903	0.853	0.875	0.797	0.738	0.832

Implications

- Text embeddings from transformer based models are best at grouping similar pieces of user feedback together

- Out of these models, USE is the most performant

Future work

- Apply these embeddings to clustering algorithms - explore which clustering algos are most suitable for RE

- Repeat work with non-English languages

AIRE 2021 questions

How does our work help the field of AI?

- Novel approach using only class-labelled data to evaluate embeddings for a particular domain. This approach can be extended to any domain.

How does our work help the field of RE?

- Presents rigorous evaluation of methods to help decision making in the creation of future tools to aid RE. It can continue to be informative as new embeddings become available.

Thanks

- Email - pdev438@aucklanduni.ac.nz

- Twitter - @p_d_research

- Replication package - https://doi.org/10.5281/zenodo.5183351

Word embedding references

1. Jones KS. **A statistical interpretation of term specificity and its application in retrieval**. Journal of documentation. 1972.
2. Blei et al. **Latent dirichlet allocation**. the Journal of machine Learning research. 2003 Mar 1;3:993-1022.
3. Yan et al. **A biterm topic model for short texts.** InProceedings of the 22nd international conference on World Wide Web 2013 May 13 (pp. 1445-1456).
4. Yin & Wang **A dirichlet multinomial mixture model-based approach for short text clustering.** KDD 2014 Aug 24 (pp. 233-242).
5. Pennington et al. **Glove: Global vectors for word representation.** EMNLP 2014 Oct (pp. 1532-1543).
6. Komninos & Manandhar **Dependency based embeddings for sentence classification tasks.** NAACL human language technologies 2016 Jun (pp. 1490-1500).
7. Ethayarajh **Unsupervised random walk sentence embeddings: A strong but simple baseline**. RepL4NLP 2018 Jul (pp. 91-100).
8. Levy & Goldberg **Dependency-based word embeddings**. ACL (Volume 2: Short Papers) 2014 Jun (pp. 302-308).
9. Reimers & Gurevych **Sentence-bert: Sentence embeddings using siamese bert-networks**. arXiv preprint arXiv:1908.10084. 2019 Aug 27.
10. Cer et al. **Universal sentence encoder for English.** EMNLP: System Demonstrations 2018 Nov (pp. 169-174).
11. Feng et al. **Language-agnostic bert sentence embedding.** arXiv preprint arXiv:2007.01852. 2020 Jul 3.

User feedback references

1. Pagano et al. **User feedback in the appstore: An empirical study**. RE 2013 Jul 15 (pp. 125-134)
2. Chen et al. **AR-miner: mining informative reviews for developers from mobile app marketplace**. ICSE 2014 May 31 (pp. 767-778)
3. Di Sorbo et al. **What would users change in my app? summarizing app reviews for recommending software changes**. FSE 2016 Nov 1 (pp. 499-510)
4. Guzman et al. **A little bird told me: Mining tweets for requirements and software evolution**. RE 2017 Sep 4 (pp. 11-20)
5. Nayebi et al. **App store mining is not enough for app improvement**. Empirical Software Engineering. 2018 Oct;23(5):2764-94
6. Williams et al. **Mining twitter feeds for software user requirements**. RE 2017 Sep 4 (pp. 1-10)
7. Tizard et al. **Can a conversation paint a picture? mining requirements in software forums**. RE 2019 Sep 23 (pp. 17-27)
8. Ali Khan et al. **Conceptualising, extracting and analysing requirements arguments in users' forums: The CrowdRE-Arg framework**. Journal of Software: Evolution and Process. 2020 Dec;32(12):e2309.