Fine-Grained Mobile Application Clustering Model Using Retrofitted Document Embedding

Authors: Yeo-Chan Yoon, Junwoo Lee, et. al.

Presented by: S.Samyukth

Overview of the past 1 week

- Briefly read and understood the concept of topic modelling and document clustering by reading research papers. (mentioned in Literature Review) and further studied the paper on Mobile Application Clustering using a proposed retro-fitted model the presentation to which commences from the next slide
- For future motivation I look to dive deeper into the coding and algorithmic part of the retrofitted proposed model of this journal and implement an efficient mobile clustering model
- I further look to solve a few limitations from this current journal (mentioned in the last few slides) using alternative better models suggested in some of the papers I've recently read (mentioned in Literature Review), I believe this new method will be of extreme value to Samsung Research for efficient document clustering.

Introduction (Purpose)

- With almost 2 million applications in play store and app store combined there is a need to cluster and classify them. The current classification in market is coarse grain and less accurate
- For example, both scanning and job-search applications are generally classified into the business category, whereas real-estate applications providing similar functions are classified into lifestyle, finance, or business categories according to the developer's determination
- The journal proposes an ideal model to classify the applications in a fine grain manner for unsupervised cluster of applications

Introduction (Summary of Research)

- Proposes a fine-grained and semi-supervised clustering method for a very large mobile application set
- Proposes a method for learning a document vector using an automatically constructed training set
- Use both word embedding and document embedding to merge similar clusters
- Compares clustering performance of proposed model (rdoc2vec) with baseline algorithms (k means, LDA, TF, RCF, doc2vec) based on purity and and entropy
- Compares classification performance of rd2v with other algorithms based on Mean Average Precision (MAP)

Literature Review

- Lin Liu et. al. “**An overview of topic modeling and its current applications in bioinformatics**” – Springer; to get a brief understanding of topic modeling
- Rubayyi Alghamdi et. al. “**A Survey of Topic Modeling in Text Mining**”- International Journal of Advanced Computer Science and Applications; to get a brief understanding of LDA and LSA
- Jiwei Li et. al. “**Do Multi-Sense Embeddings Improve Natural Language Understanding**?” – Stanford; to find an solution to one of the drawbacks in the current journal
- Guang –He Lee et. al. “**MUSE: Modularizing Unsupervised Sense Embeddings**” – MIT; to look for alternative more efficient way to solve polysemy in clusters

Problems that the journal solves

- With rise in number of mobile applications and less labels to classify them accurately, there grows a dilemma to categorize the application wherein fine granular hierarchical classification is required to better understand user behavior and activities
- For example: decathlon, H&M, reliance trends are all classified in shopping but decathlon is also an application for sport enthusiasts
- Clustering a large mobile application set with labels is very difficult
- Merging clusters (making broad categories from fine categories) requires resource construction cost for a synonym dictionary

Hypothesis

- If the proposed retrofitted document embedding model is able to automatically determine the clusters and their frequency in an unsupervised environment (no predefined categories)
- then the proposed model is able to perform fine grain clustering by initializing clusters based on title keywords and then merging similar clusters
- which will increase the purity and decrease the entropy of the process with respect to standard known algorithms

Methodology

- 2.1 million mobile applications were crawled from the Google Play Store and 1,000 mobile applications were selected for each of the following five categories: lifestyle (Life), education (Edu), travel and local (Travel), tools (Tool), and entertainment (Ent)
- Then, three annotators manually classified 5,000 mobile applications with a fine-grained hierarchical classification structure

Methodology

Initializing Clusters with Title Keywords

- The keyword selector extracts keywords as the initial cluster labels from the mobile application titles
- The extracted keywords are used to classify the applications, including the keywords in the title, into the initial clusters
- In order to improve the document embedding, the proposed model utilizes these initial clustering results as the cluster tagged training corpus

Methodology

Expanding Clusters with Document Embedding

- By using document embedding, the cluster expander allows the initial clusters to include more mobile applications without any title keywords
- It analyses title and description both
- It calculates the cosine similarity between the new cluster centroids and the document vector of every mobile application with no title keyword
- Finally, if an app doesn’t have the purpose it serves in it’s title, it’ll still be clustered in it’s respective label using the retrofitted embedding

Methodology

Merging Clusters with Word and Document Embedding

- the cluster merger combines similar clusters by using word and document embedding
- When the linear combination of label similarity, cluster similarity and cluster overlapping exceeds threshold value, similar clusters are merged

Data

Table shows the details of the test set and the crawled set. To learn the document embedding, we exploited the crawled set as the training corpus, which was automatically clustered using the cluster initializer of the proposed model

		Life	Edu	Travel	Tool	Ent
Test set	# Apps	802	806	890	840	760
	# Class	67	35	19	52	30
	# Class (top level)	43	23	15	38	24
Crawled set	# Apps	79,122	96,393	42,858	86,784	91,564
	# Class	3,865	4,399	2,019	3,302	4,411

Data

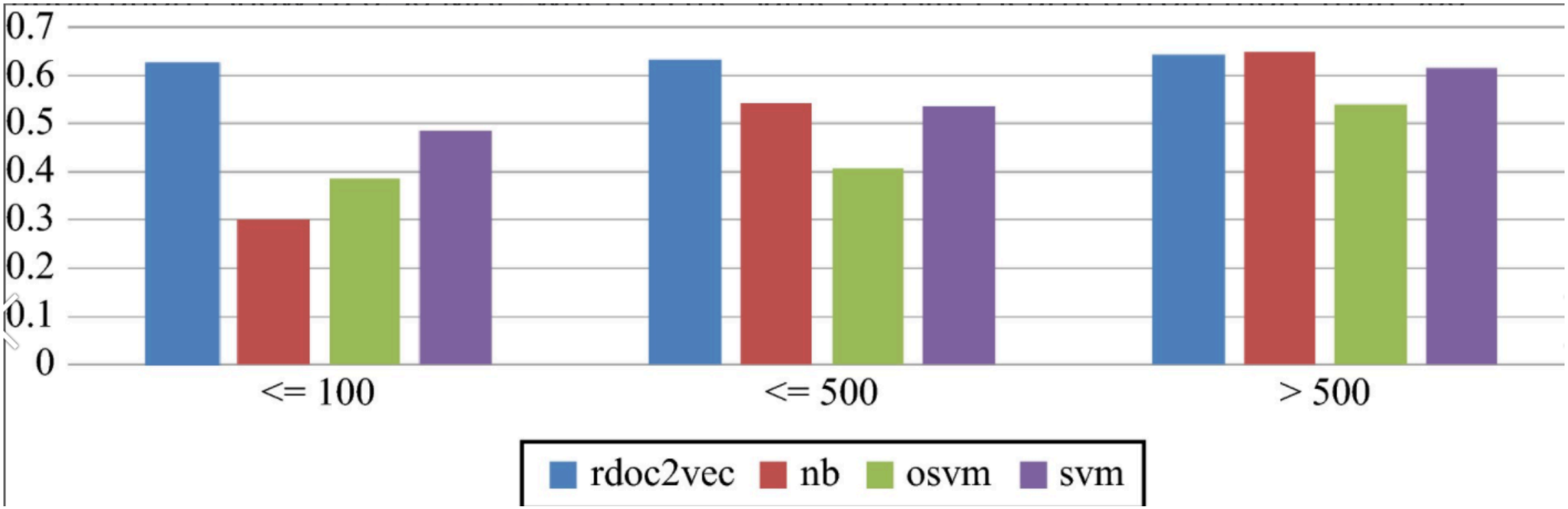
Table shows the top-five nearest neighbour mobile applications for each of the three mobile applications. The bold type indicates truly similar mobile applications, which indicates that the proposed retrofitted document embedding method *rdoc2vec* can find more truly similar mobile applications than the conventional document embedding method *doc2vec*

App	Similar apps (doc2vec)	Similar apps (rdoc2vec)
Horoscope	• Strobe Light • Living Room • Easy to Use Calculator • Love Calculator • **My Daily Horoscope Positive**	• **Horoscope** • **Daily Horoscope** • **The Horoscope** • **Horoscope** • **Cancer Horoscope**

Data

Figure shows that the MAP of the classifiers depends on the training corpus size

The proposed *rdoc2vec* method shows a stable performance regardless of the training corpus size



Discussion

Clustering Performance

- Clustering performance is compared between the proposed model and the baseline algorithms
- Evaluation measures: purity and entropy
- Output: Purity increases, Entropy decreases
- Result: the proposed model has better efficiency than baseline algorithms

Discussion

Classification Performance

- The proposed model uses an initial labeled cluster, uses it as a training set and classifies documents by comparing their similarity with existing clusters
- Evaluation measure: Mean Average Precision (MAP)
- Output: shows the best MAP for the average of five use case categories. It improves by more than 0.09 in MAP in a comparison with the three classifiers: naive Bayes, the one-class SVM, and SVM
- Result: the proposed model is more suitable for the fine-grained classification problem than a classifier

Conclusion

- The hypothesis passed and the proposed model (retrofitted doc2vec) was able to accomplish in fine grain clustering of a large dataset of applications using a training set with a high efficiency and accuracy as compared to standard algorithms
- The proposed model initializes the clusters based on the title keywords with less noise. The initialized clusters are used as a training set to learn the document embedding
- The proposed method was able to perform with high efficiency by adding a constraint to the object function. (i.e rd2v proved to better the conventional d2v)

Conclusion

- Label similarity, cluster similarity and cluster overlap were used as measures to merge similar clusters saving costs of constructing costly resources
- The journal seeks to apply the proposed fine-grained mobile-application clustering model for retrofitting the document embedding to other domains, such as an emotional analysis.

Presenter’s views

- One of the greatest limitations of the journal that I observed is the proposed model cannot take care of polysemy (same keyword with multiple meanings) of keywords.
- For example, if there is an app related to ‘river banks’ and an app related to financial banks, both of them will by default be categorized in finance by the model from the keyword ‘banks’
- The model (CBOW/ Skip-gram) may not efficiently identify combined word phrases such as New Delhi, this will be read separately by the model as ‘New’ and ‘Delhi’

Presenter’s proposed solution

- To solve this problem of polysemy, Sense embedding can be used. While in case of word embedding we used to represent a single word with a single vector, with sense embedding, we represent even a single word with different word vectors based on the sense it means

Word Vector Embedding	Sense Vector Embedding
a. tie (1 word, 5 meaning) is 1 word ~ 1 vector representation	a. tie (1 word, 5 sense (meaning)) is 1 word (5 sense) ~ 5 vector representation

For a word tie that may mean multiple things

Presenter’s proposed solution

- Sense embedding can be achieved in following ways:
- a. Clustering based word Sense Representation
  b. Non-parametric word Sense Representation
  c. Ontology- based word sense representation
- the polysemy may differ in a way that one word has 3 meanings and another has 5 but the clustering model only learns fixed number of sense for each polysemous word)
- The ability to learn a new word’s sense if it is introduced is also important (Ontology cannot handle this since it’s polysemous word senses are pre defined)
- **Solution? Non-Parametric word sense representation!**

Presenter’s concluding ideals

- I further believe that Hierarchical Dirichlet Process (assumes infinite set of distributions) can prove to be a more efficient performer than the standard LDA or LSA (assumes only fixed and finite set of distributions)
- Further, Chinese Restaurant Process (as mentioned in one of the Stanford papers) can model many kinds of dependencies between data in infinite clustering models, there could be immense possibilities to explore and achieve using ‘Chinese Restaurant Process’

Thank you