Lightweight Document Classification for Device-based APP-Recommendation: A Graph-based Approach

ABSTRACT

We consider the problem of lift document classification on mobile devices. Document classification is the task of automatically assigning a set of unlabeled documents into a set of predefined categories. This technique is relevant for app-recommendation systems on mobile phones. While app-stores provide basic recommendation functionality, more advanced recommendation systems require fine-grained usage information available only locally on the mobile device. However, due to severe resource restrictions on such devices, computational cost needs to be optimised. In this paper, summarization as an approach to circumvent the curse of dimensionality is investigated. High dimensional feature space can be reduced significantly by considering summarized document as a feature set, since it includes the most important information of the original document. Graph-based summarization technique is applied on the classification process, and remarkably improves the performance of document classification.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous; See http://acm.org/about/class/1998/ for the full list of ACM classifiers. This section is required.

Author Keywords

App recommendation; Document classification; Summarization

INTRODUCTION

It has become easy to find an app for virtually any possible category but challenging to identify good and reliable apps from this overwhelming choice. Although app-stores typically provide basic recommendation functionality, such systems favor apps with a bigger crowd of users such as corporation-developed apps or older and therefore better known apps. They can not take into account the individual interest of users and their usage habits. This problem has been tackled by app recommendation systems which require local installation [20, 17]. However, these systems are highly resource demanding and therefore not applicable in practical everyday use. What is required is a computationally cheap approach that is feasible for the application on end-user mobile devices. In this paper, we tackle this problem by considering summarization as an approach to circumvent the curse of dimensionality in document classification.

Automatic document classification (also known as text categorization, or topic spotting) is the task of automatically sorting a set of documents into categories (or classes, or topics) from a predefined set [16]. For app-recommendation systems, document classification is an essential component to group apps into categories according to their textual description, crawled, for instance, from online-appstores.

In document classification, one document is often represented as a vector of words (bag of words), and all these words are not that informative to be included in the final feature set. Therefore feature selection should be applied not only to select the most relevant features, but to reduce the high dimensionality of feature vector space. In this paper, text summarization will be considered as a feature selection technique to extract the least number of features with the most informativeness for each category.

Online app-stores and the description of apps therein are subject to constant change. Furthermore, a ground truth for correct classification is naturally missing. In order to produce comparable results and to reliably measure the performance of our approach, we apply our approach to the Reuters-21578 corpus which is a standard benchmark for document classification. It has been employed in multiple scientific publications in many research areas especially in information retrieval, natural language processing and machine learning. The hidden semantic relationship between some categories and the skewed distribution of documents make Reuters-21578 corpus most interesting for document classification with respect to app recommendation systems [5]. Moreover, it has several categories which own very few positive training examples; challenging the performance of the document classification system based on machine learning methods. The documents refer to the Reuters newswire in 1987 and the classification was done manually by personnel from Reuters Ltd. Due to its large number of categories, different subsets of its categories have been adopted as dataset. A subset of 30 categories will be taken into account for this project, with at least one positive training example and one test example.

The rest of this document is orgainzed as follows. In Section , related work is reviewed. Section presents our approach. Section details our results and section concludes the discussion.

RELATED WORK

Application recommenders have started to become increasingly commonplace, with several academic [20, 17] and commercial systems [1] emerging in recent years. Some of these operate as separate applications that are installed on the device, such as Aptoide and Cydia, but some, like Aptoide, can also be reached through a web browser on another device. At the

[1] The Aptoide meta-store: http://m.aptoide.com/

same time, recommendations have started to emerge on the marketplaces themselves, e.g. Google Play supports both personalized recommendations and country-specific "featured" and most popular application listings.

First works on mobile app recommendations were focused on adopting standard content-based and collaborative filtering techniques for generating recommendations. Most of these works operated directly on the marketplace and relied on application popularity, such as installation counts and click stream data, or ratings to generate recommendations [3]. However, as shown by Falaki et al. [6], installation counts are a poor indicator of user interest as users tend to try out applications without necessarily ever using them again. Moreover, some users may not bother uninstalling apps but rather keep apps that have been tried only once. These apps will then receive the same weight as those that are used regularly. The same holds for ratings which do not necessarily reflect true user interest. For example, many users give a one star rating for apps that do not function properly on their device [20]. At the same time, usage patterns have been shown to be highly contextualized, with many applications only being used in specific contexts [2].

Motivated by the findings of the application usage studies, most recent works on app recommendation rely on usage information gathered directly from the device. For example, AppJoy [20] considers a weighted model where recency, frequency, and duration of interactions are taken into consideration, whereas GetJar [17] and the Djinn system of Karatzoglou et al. [8] consider information derived from binary usage patterns. AppJoy relies on a constantly running background process that monitors app use, while both GetJar and our technique can be used with crowdsourced, infrequently sampled data. Both AppJoy and GetJar are based on standard recommender system techniques, whereas Djinn is based on tensor-factorization model that additionally considers also the usage context of applications. Also other works on integrating context information as part of app recommendations have been proposed [14, 4, 18]. Finally, the AppAware system of Girardello and Michahelles [7] provides a Twitter-style social media based systems where users can receive app recommendations from friends and collectively nearby users.

Usage trends of mobile applications have been monitored by using download counts [15] or in case of usage, only focusing on short period of time [2]. Recently, the AppTrends approach proposed to consider actual usage data and to base the recommendation on frequency of co-usage of apps [1]. The show that recommendations should consider co-usage of apps as particular use cases on mobile devices involve a common set of apps rather than individual apps only.

As it is mentioned before, feature selection as a fundamental task plays an important role in the overall performance of automatic DC. Many techniques and approaches has been studied and deployed in feature selection, which all focus on aggressive dimensionality reduction. Apart from common feature selection methods such as Document Frequency (DF), Information Gain (IG), Statistic (CHI) and Mutual Information (MI), in several researches text summarization was applied as feature selection and it was found useful and beneficial in automatic DC. Ker and Chen [9] proposed a summarization-based document classification system. Among Several techniques for text summarization (which includes methods based on position, cue phrase, word frequency and discourse segmentation) word-based frequency and position methods were considered and then combined to extract features. From position point of view, title of the document was only used with the assumption that existing words in the title probably describes the context relatively well. After weighting the selected features, DC process was performed by a probabilistic classifier runs on TF-IDF (Term frequencyâĂŞInverse Document Frequency). The experiment showed that using title as a summarization technique would result in acceptable performance, meanwhile decline the execution time.

The work by Kolcz et al. [12] tried to prove the efficiency of their proposed summarization technique by using it as a feature selection method in DC. Different summarization methods based on the title of the story, paragraphs and best sentence were considered in the approach in order to reduce the feature set to a manageable size. Paragraphs' position, keywords and title words were taken into account in summary generation, which included first paragraph, first two paragraph, first last paragraph, paragraph with most keywords, paragraph with most title words. In addition, another summary was also constructed by choosing the sentences with at least 3 title words and at least 4 keywords. The applied classifier was Support Vector Machine (SVM). The applied summarization methods were as effective as state-of-art statistical feature selection methods in DC, specially the best sentence-based summary. In another text summarization method, Ko et al. [11] determined the importance of sentences in a document by combining two methods. First, instead of directly using terms of the title, the most similar sentences with the title were selected, and then in the second method the sentence with the highest sum of weighted terms (based on TF, IDF, and X2 statistics values). Then, the chosen sentences were scored by a modified weighting technique, to retrieve the most informative sentence. The suggested system enhanced the performance of document classification in four applied classifiers: Naive Bayes, Rocchio, k-NN, and SVM classifier, regardless of specific language. Mihalcea and Hassan [13] presented a new approach by summarizing the documents in order to improve and enhance the classifier which results in efficient execution of a DC task. The extraction of sentences was performed with the aid of graph-based algorithms which ranked them according to the number of links. Two popular ranking algorithms: PageRank citeBrin20123825 and HITS (Hyperlinked Induced Topic Search) [10] were deployed in order to decide on the importance of sentences which finally construct the summarization. Mapping these algorithms to the document, each sentence is considered as a vertice, and if each pair of sentences have some informative terms in common, then there will be an edge between them. Naive Bayes, Rocchio were applied as main classifiers. The results revealed that the proposed system improved the classification efficiency, disconsidering the length of the original document. The technique was also recommended as a measurement for different summarization tool evaluation. Jiang et al. [19] examined the impact of summarzation on document classification, by considering two different applications of summary in order to obtain the feature set: considering the summary itself as the feature set and applying classical feature selection method (MI, IG and DF) on summary. To construct the summary, only nouns and verbs were weighted, with regards to semantic-based distance value and connective strength value. The calculated weights then were used in determining the constituent sentences of the final summarization. The outcome indicated that the approach decreased the classification computations and provided a high speed system with acceptable performance.

The reviewed approaches and methods in this section present different methods of summarization. Initial works considered the summary as a part or fraction of the document (title, paragraph, etc.). Later works try to generate more precise summaries, using many complicated term-based and sentence-based weighting techniques. Graph-based algorithms were also quite successful in creating valuable summaries. Many informative features were missing in the simple extraction techniques, for instance title itself cannot be a good candidate for a summary, since there is at least one sentence in a document which is more descriptive. On the contrary, complex summarization methods lead in more reliable summaries, since many aspects are taken into account in their process. However, summarization as a feature selection method is a promising technique to effectively improve the performance of document classification systems.

METHODOLOGY

The styles contained in this document have been modified from the default styles to reflect ACM formatting conventions. For example, content paragraphs like this one are formatted using the Normal style.

Preprocessing

Your paper's title, authors and affiliations should run across the full width of the page in a single column 17.8 cm (7 in.) wide. The title should be in Helvetica or Arial 18-point bold. Authors' names should be in Times New Roman or Times Roman 12-point bold, and affiliations in 12-point regular.

See `\author` section of this template for instructions on how to format the authors. For more than three authors, you may have to place some address information in a footnote, or in a named section at the end of your paper. Names may optionally be placed in a single centered row instead of at the top of each column. Leave one 10-point line of white space below the last line of affiliations.

Document Summarization

Every submission should begin with an abstract of about 150 words, followed by a set of Author Keywords and ACM Classification Keywords. The abstract and keywords should be placed in the left column of the first page under the left half of the title. The abstract should be a concise statement of the problem, approach, and conclusions of the work described. It should clearly state the paper's contribution to the field of HCI.

Classification

Please use a 10-point Times New Roman or Times Roman font or, if this is unavailable, another proportional font with serifs, as close as possible in appearance to Times Roman 10-point. Other than Helvetica or Arial headings, please use sans-serif or non-proportional fonts only for special purposes, such as source code text.

EXPERIMENTS AND RESULTS

The heading of a section should be in Helvetica or Arial 9-point bold, all in capitals. Sections should *not* be numbered.

CONCLUSION

Place figures and tables at the top or bottom of the appropriate column or columns, on the same page as the relevant text (see Figure **??**). A figure or table may extend across both columns to a maximum width of 17.78 cm (7 in.).

REFERENCES

1. Donghwan Bae, Keejun Han, J. Park, and M.Y. Yi. 2015. AppTrends: A graph-based mobile app recommendation system using usage history. In *Big Data and Smart Computing (BigComp), 2015 International Conference on*. 210–216. DOI: http://dx.doi.org/10.1109/35021BIGCOMP.2015.7072833

2. Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 47–56. DOI:http://dx.doi.org/10.1145/2037373.2037383

3. Fredrik Boström, Petteri Nurmi, Patrik Floréen, Tianyan Liu, Tiina-Kaisa Oikarinen, Akos Vetek, and Péter Boda. 2008. Capricorn - an Intelligent User Interface for Mobile Widgets. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. ACM, New York, NY, USA, 327–330. DOI: http://dx.doi.org/10.1145/1409240.1409280

4. Christoffer Davidsson and Simon Moritz. 2011. Utilizing Implicit Feedback and Context to Recommend Mobile Applications from First Use. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation (CaRR '11)*. ACM, New York, NY, USA, 19–22. DOI: http://dx.doi.org/10.1145/1961634.1961639

5. Franca Debole and Fabrizio Sebastiani. 2005. An analysis of the relative hardness of Reuters-21578 subsets. *Journal of the American Society for Information Science and technology* 56, 6 (2005), 584–596. http://onlinelibrary.wiley.com/doi/10.1002/asi.20147/full

6. Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. 2010. Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on*

Mobile Systems, Applications, and Services (MobiSys '10). ACM, New York, NY, USA, 179–194. DOI: http://dx.doi.org/10.1145/1814433.1814453

7. Andrea Girardello and Florian Michahelles. 2010. AppAware: Which Mobile Applications Are Hot?. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 431–434. DOI: http://dx.doi.org/10.1145/1851600.1851698

8. Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. 2012. Climbing the App Wall: Enabling Mobile App Discovery Through Context-aware Recommendations. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. ACM, New York, NY, USA, 2527–2530. DOI: http://dx.doi.org/10.1145/2396761.2398683

9. Sue J. Ker and Jen-Nan Chen. 2000. A Text Categorization Based on Summarization Technique. In *Proceedings of the ACL-2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 11 (RANLPIR '00)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 79–83. DOI:http://dx.doi.org/10.3115/1117755.1117766

10. Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46, 5 (Sept. 1999), 604–632. DOI:http://dx.doi.org/10.1145/324133.324140

11. Youngjoong Ko, Jinwoo Park, and Jungyun Seo. 2004. Improving Text Categorization Using the Importance of Sentences. *Inf. Process. Manage.* 40, 1 (Jan. 2004), 65–79. DOI:http://dx.doi.org/10.1016/S0306-4573(02)00056-0

12. Aleksander Kolcz, Vidya Prabakarmurthi, and Jugal Kalita. 2001. Summarization As Feature Selection for Text Categorization. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM '01)*. ACM, New York, NY, USA, 365–370. DOI:http://dx.doi.org/10.1145/502585.502647

13. Rada Mihalcea and Samer Hassan. 2005. Using the essence of texts to improve document classification. (2005). http://digital.library.unt.edu/ark:/67531/metadc30978/

14. Stefano Mizzaro, Marco Pavan, Ivan Scagnetto, and Ivano Zanello. 2014. A Context-aware Retrieval System for Mobile Applications. In *Proceedings of the 4th Workshop on Context-Awareness in Retrieval and Recommendation (CARR '14)*. ACM, New York, NY, USA, 18–25. DOI: http://dx.doi.org/10.1145/2601301.2601305

15. Thanasis Petsas, Antonis Papadogiannakis, Michalis Polychronakis, Evangelos P. Markatos, and Thomas Karagiannis. 2013. Rise of the Planet of the Apps: A Systematic Study of the Mobile App Ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC '13)*. ACM, New York, NY, USA, 277–290. DOI: http://dx.doi.org/10.1145/2504730.2504749

16. Fabrizio Sebastiani. 2002. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.* 34, 1 (March 2002), 1–47. DOI: http://dx.doi.org/10.1145/505282.505283

17. Kent Shi and Kamal Ali. 2012. GetJar Mobile Application Recommendations with Very Sparse Datasets. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 204–212. DOI: http://dx.doi.org/10.1145/2339530.2339563

18. Wolfgang Woerndl, C. Schueller, and R. Wojtech. 2007. A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. 871–878. DOI: http://dx.doi.org/10.1109/ICDEW.2007.4401078

19. Jiang Xiao-Yu, Fan Xiao-Zhong, Wang Zhi-Fei, and Jia Ke-Liang. 2009. Improving the Performance of Text Categorization Using Automatic Summarization. In *Computer Modeling and Simulation, 2009. ICCMS '09. International Conference on*. 347–351. DOI: http://dx.doi.org/10.1109/ICCMS.2009.29

20. Bo Yan and Guanling Chen. 2011. AppJoy: Personalized Mobile Application Discovery. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*. ACM, New York, NY, USA, 113–126. DOI: http://dx.doi.org/10.1145/1999995.2000007