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 [19, 16]. 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 [15]. 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 [6]. 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 [19, 16] 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. [7], installation counts are a poor indicator of user interest as users tend to try out applications without necessarily ever using them again. The same holds for ratings which do not necessarily reflect true user interest [19]. At the same time, user habits and personalised usage patterns have a high potential to significantly improve app recommendation systems [2].

Motivated by these findings, most recent work on app recommendation relies on usage information gathered directly from the device. For example, AppJoy [19] considers a weighted model where recency, frequency, and duration of interactions are taken into consideration, whereas GetJar [16] 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, 5, 17].

Recently, the AppTrends approach proposed to consider actual usage data and to base the recommendation on frequency of co-usage of apps [1]. They 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. While these systems can provide better and more personalised recommendation results than the basic approaches provided via app stores, the high resource demands of the mobile-based solutions render practical use impossible. We therefore investigate lightweight document classification to mitigate this problem.

Feature selection as a fundamental task plays an important role in the overall performance of automatic document classification. 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 and Mutual Information (MI), in several researches text summarization was applied as feature selection and it was found useful and beneficial in automatic document classification.

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 likely describe the context relatively well. After weighting the selected features, document classification was performed by a probabilistic classifier exploring 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 document classification. 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 and 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 a 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). Afterwards, 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 document classification. 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 [4] 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. [18] examined the impact of summarzation on document classification, by considering two different applications of summary in order to obtain the feature set: the summary itself as the feature set and applying classical feature selection methods (MI, IG and DF) on the summary. To construct the summary, only nouns and verbs were weighted, with regard to semantic-based distance value and connective strength value. The calculated weights were then 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.

Many informative features were missing in the above mentioned simple extraction techniques. For instance, the title itself, especially in app recommendation systems, cannot be a good candidate for a summary. On the contrary, the more advanced summarization methods lead to reliable summaries, since many aspects are taken into account in their process. Summarization as a feature selection method is a promising technique to effectively improve the performance of document classification for app recommendation systems.

METHODOLOGY

The proposed DC system utilizes the advantages of using summary instead of the complete document, since it includes the standpoints and main information in a shorter format. The training phase starts with text preprocessing which involves tokenization, stopword removal and stemming. To improve the quality of features, WordNet and porter stemmer are applied in this step. The documents will be then summarized by a graph-based text summarization approach proposed by [10],in which sentences constructed the summary are chosen by shortest path algorithms. After replacing the documents with their related summaries, a specific feature set for each category will be determined by considering the words of summarized documents as the final feature set. K-Nearest Neighbor (KNN) algorithm is used in the test phase to classify unlabeled documents in the test set and assign each to its relevant category.

Preprocessing

The performance of ML is highly influenced by data preprocessing, and the quality of the input data determines the quality of output data. Mapping this fact to the DC process, an efficient preprocessing technique should be employed to evaluate the words (features) and choose the dominant ones to be used later in classifier training phase. Irrelevant and redundant words make it difficult for the classifier to decide. Morphological analysis is a productive step in the quality improvement of input textual data, which is divided into tokenization, stemming and recognition of ending of records [11]. During the tokenization, a dataset of meaningful words (tokens) - separated by a special âĂŸspaceâĂŹ character - will be produced and other tokens such as punctuation, numbers etc. will be eliminated. Stemming is the process of transforming the inflections of words into their related stem (root) [12]. Many automatic stemming approaches have been proposed like successor variety stemmer, n-gram method and affix removal stemmer. The most widely used of all the stemmers is the porter stemmer, which was first presented in 1980. The stemmer is a context sensitive suffix removal algorithm and is available in many languages. It utilizes several sequential steps - five or six- to find the stem. The improvement of performance is expected with regards to the fact that stemming boosts the precision and recall in IR systems. Finally, documents' segments (e.g. sentence, paragraph) are also be recognized by searching end of records like '.','!'. Detected segments later will be used as desired units [11]. WordNet [13] is a useful tool for computational linguistics and natural language processing (NLP), because of its well- designed structure. The synonymy in WordNet is an important semantic relation between words, for example shut and close or car and automobile. This english-based lexical database, uses the concept of synset which is a group of nouns, verbs, adjectives and adverbs cognitively synonyms. Synsets share the same concept in many contexts, are grouped in WordNet. In addition, each pair of grouped synsets are linked by using conceptual relations [14]. The selection of words from the synonymsâĂŹ perspective, is performed with regard to the co- occurrence of terms in a category which belongs to the same synset signature. The synset signature cross-referencing has the main role in selecting the correct sense for the terms co- occurrence in a category. The process is performed by examining the list of noun synsets for all senses in order to acquire the similarity in the signatures. The semantic context of a category would be formed by the senses of different words which their synsets overlap. The equal synsets with the same signatures would be added to the feature lists of the category in order to select the original terms.

Document Summarization

To reduce the high dimensionality of feature space, the documents are required to be shortened while preserving the key points. Since summary is a short version of a document, Summarization as an effective method contributes to the objective. A set of words, sentences or paragraphs - derived of a text document or a collection of documents - is called a summary if it points to the same information or concepts as the original one(s). Two common methods for generating a summary are abstractive summarization and extractive summarization. The summary produced by the first method, is mostly semantic-based and might not contain exact original words. In contrast, extractive methods create the summary made of genuine words of the reference document(s). In this work extractive summarization is considered which is strongly associated to feature selection, due to straight forward generation of a summary. The adopted graph-based summarization method in this work, is suggested by [10]. To represent the original document in the graph format, first it is divided into sentences. Each sentence is now considered as a vertice, and there will be a weighted edge between each pair of sentences, if they share one word at least. In addition to the relation edge, each sentence is also connected to the following sentence with an unweighted edge to facilitate the navigation through the graph. The assigned cost to each edge denotes high similarity, in case of low cost and vice versa. The cost of edge from sentence i (Si) to the sentence j (Sj) is calculated as:

Overlapi,j returns the number of words shared between sentence i and j. In the proposed summarization only words of four characters or more are considered, but in our approach this condition is discarded. The reason is that stemming reduces the words into their roots, so the number of characters within a word will also be reduced, and this may cause inaccuracy. When the graph is built, the next step would be to generate the summary with the shortest path approach deployment, in which the target node corresponds to the last sentence in the main document and the source node is the first sentence. The traverse starts at the source node, and then all the cost values of the out coming edges are added to a priority queue. The path with the minimum cost will be discovered , as soon as it ends with the target node. The summary then will be created by existing sentences in this path. All the summarized documents of each category, then make a feature set in the bag-of-words format.

Classification

C. Classification KNN [15] is a classical statistical pattern recognition algorithm. Its idea is very simple: Similarity between a text document to be classified and all train samples in each category will be calculated according to a similarity computing strategy. Finally it will be assigned to the category with the maximum sum of similarities. After feature selection phase, there is only one document in each train set category which is actually our feature set. In fact, the similarity has to be computed only between the test document and just one train sample.

Where x is the document to be classified, fi is the ith feature set, ci is the ith category, and y(fi,ci) 0,1 denotes whether feature set belongs to category c i (if belongs to ci then y(fi,ci) 1, otherwise y(fi,ci) 0).

Evaluation

Precision, Recall, and F-measure are widely used among the various methods to determine the effectiveness of automatic TC. These measures are computed with regards to classification of a text. A document can be classified [16] as True Positive(TP): a document being classified correctly as relating to a category , False Positive(FP): a document that is said to be related to the category incorrectly, False Negative(FN): a document that is not marked as related to a category but should be, and True Negative(TN): a document that should not be marked as being in a particular category and is not. Both precision and recall are therefore based on an understanding and measure of relevance. Recall is the proportion of relevant material actually retrieved, and Precision is the proportion of retrieved material actually relevant. Precision and Recall are defined as follow: In order to estimate the effectiveness of the system, Precision and Recall are often combined and used in evaluation process as F-measure. In general , itâĂŹs defined as following formula: In Where P and R denote Precision and Recall respectively. By assigning 1 to Îš, F1-measure is produced, which is a harmonic mean of precision (P) and recall (R).

EXPERIMENTS AND RESULTS

Subset of thirty categories from Reuters-21578 corpus is randomly chosen as the dataset for the experiment. Number of documents in various categories is different, both in train set and test set. The documents are of different lengths from minimum one to maximum 30 sentences, considering sentence as a unit of measure for documents. So dataset includes both short and long documents. Applying the graph-based single summarization on the documents of each category, the feature set for each one is obtained. The number of features (words) in each feature set was reduced significantly, in comparison with number of features in original documents. Figure.1 indicates the reduction of dimensionality curse. Since all the categories cannot be shown, the diagram includes some random categories. For comparative evaluation, the classification was performed and measured; for both the original full-length documents and the summarized documents. The maximum and minimum number of documents for the train set was 388 and 7 respectively. As it is shown in figure 2 & 3, the precision and recall for different categories have improved noticeably using the graph-summarized approach. The classification performance accuracy determined by F1- measure shown in figure 4 indicates the great impact of the summarization; the F1-mesure has increased significantly for the summarized documents in comparison with the full length documents. The length of documents and the number of documents in the train set significantly influence the results. When one of these factors increases (or both of them) , the feature set will also increase. In case of this, the possibility of occurrence of common words between categories increases and finally leads to the reduction of Precision as a result. A relatively large feature set may increase the Recall positively, but it affects the Precision negatively.

CONCLUSION

The proposed DC system utilizes the advantages of using In this paper, we investigated the applying of a graph-based summarization approach to the document classification. The evaluation was performed for the original and summarized documents. First, we used the extractive summarization algorithm to reduce the number of features and solved the curse of dimension. Second, we showed how the extractive summarization could be effectively and usefully applied in KNN-classification for documents and improved the acquired results.

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. Sergey Brin and Lawrence Page. 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks* 56, 18 (2012), 3825 – 3833. DOI: http://dx.doi.org/10.1016/j.comnet.2012.10.007 The {WEB} we live in.
5. 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
6. 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
7. 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
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. 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
16. 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
17. 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
18. 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
19. 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