July 2022

Identifying Duplicate Apps Across Platforms

Trevor Lu

Maggie Cai

Krystal Rose Higgins

Dominick Ng

Identifying Duplicate Apps Across Platforms

ABSTRACT

App developers release versions of their apps designed for different platforms or operating systems. While there may be variations between the different app versions across platforms, they are fundamentally equivalent. However, there is currently no way of easily knowing whether an app is equivalent to another. This disclosure describes app deduplication techniques that identify apps that are duplicates. Identification of duplicate apps is performed based on app metadata using rules or machine learning models. App deduplication can improve user experience in a variety of ways, e.g., making recommendations to users discovering apps; suppressing duplicate notifications; helping users identify the right app to launch; disambiguating apps during user journeys; etc.

KEYWORDS

- Duplicate app
- Redundant app
- Deduplication
- Mobile app
- Smartphone app
- Convertible laptop
- Two-in-one laptop
- Multiplatform device
- Duplicate notification

BACKGROUND

App developers release versions of their apps designed for different platforms such as desktop, smartphone, tablet, television or other entertainment device, smartwatch or another wearable device, browser, etc. For many apps, some functionality can also be accessed through a web browser without installing the app locally on a user device. There are also devices that allow installation/execution of multiple versions of the same app, e.g., touch-screen laptops may allow both a desktop version (which may be non-touch enabled) and a tablet version (touch-enabled) of the same app. While there may be variations between the different app versions across platforms (e.g., visual differences, feature variations), they are fundamentally equivalent. The apps are branded in the same way and offer the same core product. However, there is currently no way of easily knowing whether an app is equivalent to another.

DESCRIPTION

This disclosure describes app deduplication techniques that take a known repository of apps from multiple platforms and identify apps from that repository that can be considered as duplicates. Identification of duplicate apps can be performed using any suitable technique such as a rules engine, machine learning, etc. Alternatively, duplicates can also be manually identified. A variety of app metadata inputs such as the app identifier; app name; app icon; developer name; developer identifier; known associations between apps (e.g., verified asset links between an app and a website); etc., can be used to identify duplicate apps. The metadata is used to generate a prediction over whether a given set of apps are duplicates or not.

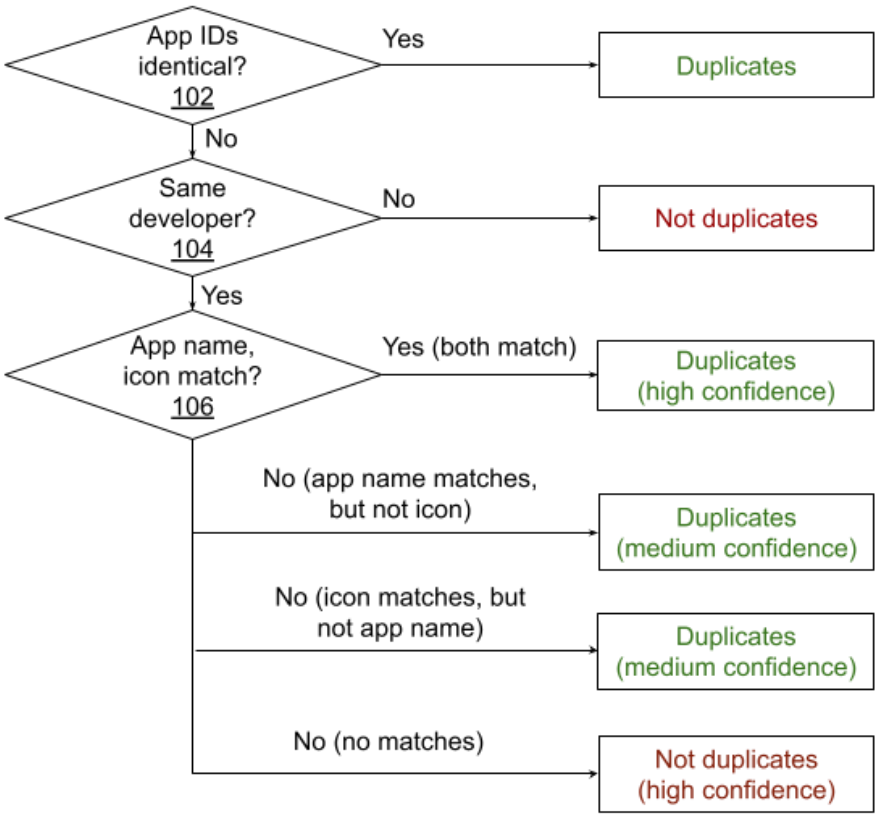

Fig. 1: Rule-based identification of duplicate apps

Fig. 1 illustrates an example of rule-based identification of duplicate apps. If the identifiers of two apps are identical (102), the apps are identified as duplicates. If the developers of two apps are distinct (104), the apps are considered non-duplicates. If the app name and/or icon match (106), then the apps are considered duplicates at varying levels of confidence.

For example, if both app name and icon match, there is a high degree of confidence that the apps are duplicate. If only one of the app name and icon match, there is a medium degree of confidence that the apps are duplicate. If neither app name nor icon match, there is a high degree of confidence that the apps are non-duplicates. App names can be matched using standard text-similarity measures, e.g., edit distance, etc. App icons can be matched using standard image-similarity techniques.

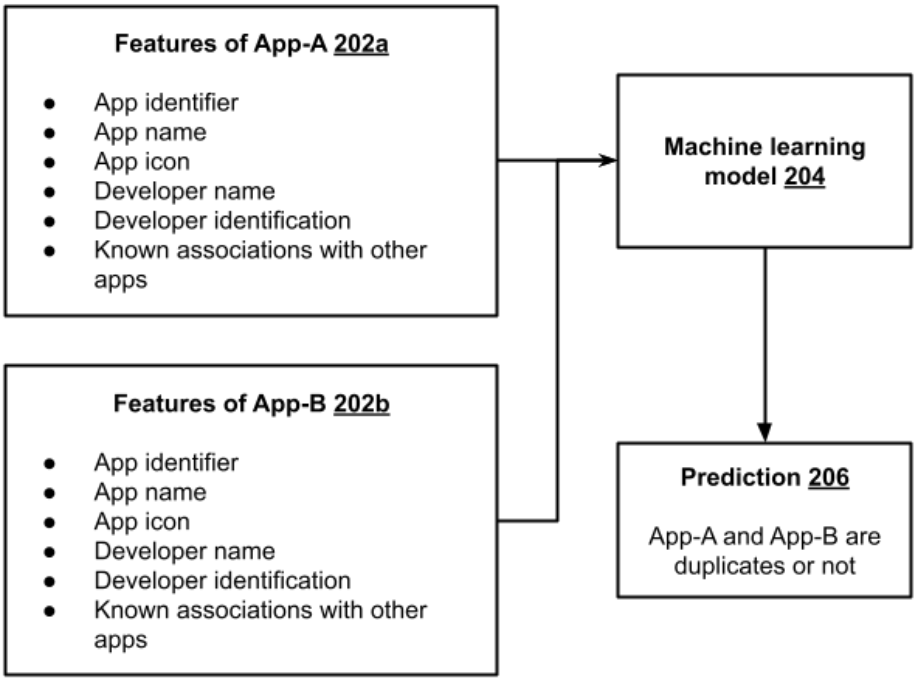

Fig. 2: App-duplicate identification using a machine learning model

Fig. 2 illustrates duplicate app identification using a machine learning model. Metadata corresponding to two apps (202a-b) are presented as input features to a machine learning model (204) that is trained to predict (206) if the two input apps are duplicates or not.

Once duplicates are identified, this information can be used to improve user experience with apps in a variety of ways, such as:

- making suitable recommendations to users when discovering apps for installation on a device;
- helping users identify which app to launch when there are duplicates available/ installed on a device;
- helping users disambiguate apps through key user journeys, e.g., app discovery; intent flow; permissions; notifications; app launching; link navigation; file handling and sharing; uninstall; etc.;

- suppressing duplicate notifications when multiple versions of the same app are installed on a device; etc.

Broadly, the described deduplication techniques reduce the chances of unnecessary duplicate apps being installed on a device; disambiguate between already installed duplicate apps and help users and enterprise IT administrators understand the differences between them to enable them to make good choices; help resolve confusing duplicate app issues for the user; help in reporting app usage metrics accurately; etc.

Duplicate notifications can be suppressed across devices such that a user simultaneously using a laptop and a mobile device, each with different versions of the same app, receives just one notification. The techniques are effective in solving the deduplication problem even in situations where the operating system developer doesn't have control over the different app ecosystems since deduplication can be done without input from the app itself (which otherwise requires the app developer to incorporate deduplication features). App deduplication can be provided in the form of a cloud service that is queried by client devices to determine if one or more apps are duplicates. Deduplication can also be implemented on client devices.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable the collection of user information (e.g., information about a user's devices and/or apps installed, or a user's preferences,), and if the user is sent content or communications from a server. In addition, certain data is treated in one or more ways before it is stored or used so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user. Thus, the user has

control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

App developers release versions of their apps designed for different platforms or operating systems. While there may be variations between the different app versions across platforms, they are fundamentally equivalent. However, there is currently no way of easily knowing whether an app is equivalent to another. This disclosure describes app deduplication techniques that identify apps that are duplicates. Identification of duplicate apps is performed based on app metadata using rules or machine learning models. App deduplication can improve user experience in a variety of ways, e.g., making recommendations to users discovering apps; suppressing duplicate notifications; helping users identify the right app to launch; disambiguating apps during user journeys; etc.