An Automated Mobile App Market Evaluation Approach

Student Numbers: 133348, 133802, 19728 and 132430

Text Analytics, 15 pages

MSc in Business Administration and Information Systems - Data Science

Supervisor: Raghava Rao Mukkamala

Copenhagen, 18.06.2020

Abstract

A market evaluation is known as a difficult and time consuming process, even though most of the unstructured data available is not included in it today and therefore holds a big potential to find growth opportunities within markets. This research aims to present an semi-automated approach to analyze descriptions of apps in the Google Play Store, by applying unsupervised methods to cluster them into groups and evaluate the clusters within a matrix, based on underlying performance measures. For that the Google Play Store was crawled to obtain the necessary information and textual similarities were identified, based on an LDA and Doc2Vec modelling approach. Both modelling results were clustered into segments with the k-means algorithm, in order to transform them into the business relevant measures, rating and number of downloads, to eventually interpret the meaning of the relative position they were mapped to. The modelling techniques were able to find textual similarities in a human-interpretable manner and revealed different levels of attractiveness for different app genres, based on the clustering. The findings of this pre-selection process helps to break down the sheer information available and reduce the necessary time to invest into a market evaluation.

Keywords: Text Analytics, Natural Language Processing, Market Research, Google Play Store, LDA, Word2Vec, Crawler, Clustering

1 Introduction (133348)

It is estimated that around 80-90% of available data is stored in an unstructured form, such as image, audio or text. However, the majority of data analyzed in the business world remains to be structure data [14]. Due to its known structure, numerical data can be analyzed with ease and achieve precise results. For textual data on the other hand, which is indispensable in every imaginable part of life, analyses remain difficult. Different factors such as context and ambiguity make it difficult to make sense out of textual data. So far, there is still a lack of sufficient tools to analyze it properly and therefore holds a huge potential to win valuable insights and enhance decision-making on the business side [13]. With the acknowledgment of the utility of text analytics in a manifold of practical use cases, over the last decades the topic received much attention. As a result, research efforts in the field of text analytics strongly increased and continue to lead to steady improvements [8].

Market analysis are still often done manually, especially with qualitative information, such as text. This remains highly time consuming, and has the risk to become outdated quickly [17]. Text analytics creates new opportunities to simplify, accelerate and enhance the process of market research in an automated way, to create a targeted product and reduce the likelihood of failure.

One market that shows continuous growth is the mobile app market. As Sensor Tower states, in 2019 Google's platform *Google Play* created a revenue of $29.4 billion. This represents an 18.8% increase compared to the previous year. Also the number of global app downloads increased by 11.4% to 84.3 billion in the Google Play Store for the year 2019. However, some markets begin to show saturation of mobile users [37]. Therefore, it is essential in app development, to consciously decide, which sector to focus on, to increase the chances of providing the customers with a desired solution and bring a successful app to the market.

Bringing all this together, the goal of this paper is to answer the question, if it is possible to structure the app market in a reasonable way to identify niche or early-stage growth markets, based on methods of the text analytics field. To do so, in the first step a crawler is build, which retrieves provided information of the apps such as ratings, descriptions and download numbers from the Google Play Store. Subsequently, the information is cleaned and different topic modelling methods, such as LDA and Doc2Vec, combined with k-means, will be applied to identify more granular categories than provided by Google's genres. In the final step these sub-groups will be evaluated in a performance-measure matrix, to identify promising markets.

2 Related Work (19728)

Main focus of the research on Google Play Store applications is identifying potential markets. Whereas companies are constantly trying to find growth markets, research concentrates in providing the companies with modern and optimized tools, as text analysis techniques, to do so. Qu et al. assessed description-to-permission fidelity of apps by applying natural language processing (NLP) processes to the Google Play store [40]. Ma, Wang, Lo, Deng and Sun combined text analytics and its various techniques with a semi-supervised approach for comparing app behaviors against its descriptions to detect malwares [26].

Fu et al. applied sentiment analysis and topic modelling to gain insight into millions of user reviews [15]. Harman et al. utilized app store mining and analysis as a form of software repository mining to extract feature information to analyze Blackberry applications technical, customer and business aspects. The findings suggested a correlation between customer rating and the rank of app downloads [18]. Our research goal is partly the extension of Harman et al. [18], but focuses on potential markets drawn from application ratings and downloads.

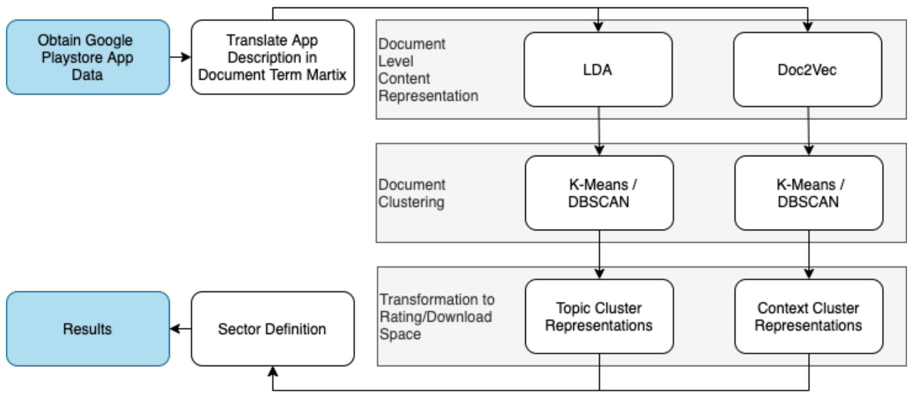
Guo et al. combined the techniques of web crawler, NLP and machine learning (ML) algorithms with data visualization to develop a big data competitor-analysis system [46]. Mokarizadeh et al. analyzed the descriptions of applications from Google Play store to provide insight concerning intrinsic properties of applications repositories. One of the findings was a high competition between app developers producing similar applications[35]. Our research goal was inspired by this finding. Structuring the Google App store in a reasonable way that provides insight into potential markets can contribute positively to app developer's decision-making process.

Bringing all this together, two main propositions for the work following in this paper are drawn:

Proposition 1: Machine learning can be used, to create a text based categorization of apps that results in a human-comprehensible outcome.

Proposition 2: Different app clusters, based on the found differences in the apps' descriptions, can be compared to each other using popularity measures, pointing out possible market opportunities in the Google Play Store market.

3 Conceptual Framework (133802)



Obtain Google Playstore App Data
Translate App Description in Document Term Martix
Document Level Content Representation
LDA
Doc2Vec
Document Clustering
K-Means / DBSCAN
K-Means / DBSCAN
Transformation to Rating/Download Space
Topic Cluster Representations
Context Cluster Representations
Sector Definition
Results

Figure 1: This visualization structures the work in its logical parts: data obtainment and preprocessing (first two steps), data analysis and manipulation (grey rectangles) and interpretation of the analysis and results (last two steps).

The total process developed in this work can be divided into four main steps: data obtainment and preprocessing, text analysis, app cluster detection and business context application. As all steps are visualized in figure 1, the work starts off by obtaining metadata related to the mobile apps. Section 4.1 therefore describes the Python program (web crawler) and how the content of each app's web page is assessed and downloaded. The downloaded data is subsequently structured and preprocessed (compare section 4.3).

Section 4.4 explains the two textual analysis algorithms, namely Latent Dirichlet Allocation (LDA) and Doc2Vec, which are applied on the app descriptions. These algorithms transform text documents into vector representations of the underlying topics, and context respectively. The generated vector representations of the app descriptions are clustered using unsupervised k-means and DBSCAN algorithms (section 4.5). As a consequence, each of the found clusters can be interpreted as a collection of similar apps based on the beforehand applied LDA and Doc2Vec models. Section 4.6 describes selected app market, which is represented by the previously found app clusters. As these clusters contain apps that hold the popularity measures rating and number of downloads, the clusters themselves can be represented by the average rating and downloads of the contained apps. Eventually, the app clusters found for each modelling algorithm, can be visualized in a two dimensional matrix.

In section 5, not only the found clusters are compared relatively to themselves, but the differences in the results of the separate machine learning algorithms are compared to each other. Thus, does chapter 5 provide the results and interpretation of the rating/download-matrix. In the final section 6, the findings of this research are discussed and concluded.

4 Methodology

4.1 Crawler (133802)

In general, a web crawler is defined as "a program that, given one or more seed URLs, downloads the web pages associated with these URLs, extracts any hyperlinks contained in them, and recursively continues to download the web pages identified by these hyperlinks" [36, p.1]. In this particular case, the crawler is build to work solely for the purposes of app data retrieval in the Google Play Store environment. As the crawler is used within a limited domain, the procedure can be visualized as shown in figure 2. Considering the structure of the online platform, the crawler works itself through three URL frontiers. First it downloads I category links from the page corresponding to the seed URL, then on each category page it downloads J_i cluster links, before it downloads the $K_{i,j}$ app pages for each cluster link added.

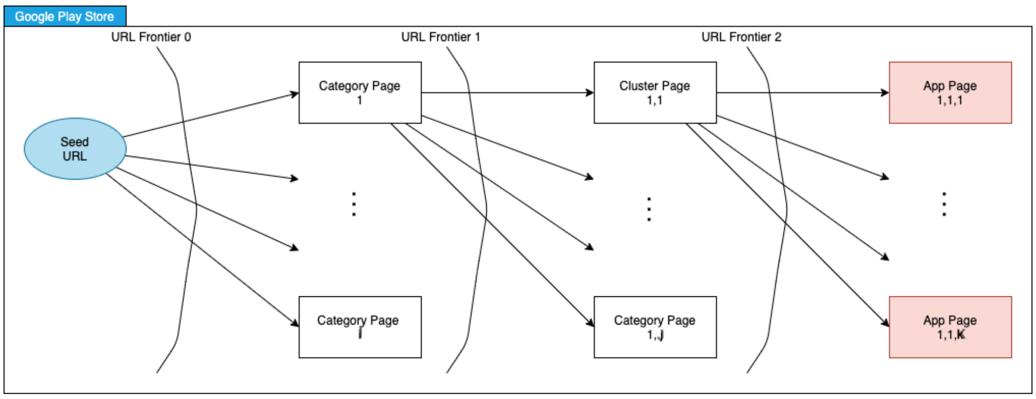


Figure 2: Schematic approach of the crawler working through three URL frontiers.

Mandatory crawler properties Because web crawlers are powerful tools to quickly access immense amounts of data from online sources, they must be both robust and polite [29]. The politeness of the crawler is ensured by only accessing pages which are marked as allowed in the https://play.google.com/robots.txt file. Also, by sending one server request per two seconds on average, the crawler does not cause a significant increase in the server's work load. The crawler's need to be robust can be considered as low, regarding the point that it only operates on a limited and a priori known environment. The program is featured with a 404 page not found detector and skips app pages which do not behave as expected. These measures are in this case sufficient for crawling without interruptions.

Optional crawler properties As suggested by Manning et al., a crawler should include more properties than only politeness and robustness. In this case, the crawler additionally allows for multi process crawling which reduces the total crawling time from about eight hours to about one and a half hours (-80%). Also, duplicated URLs are eliminated, if they appear within the same app category. A high quality of the crawled data is ensured since the app page structure is known and does not vary for different apps.

Python packages and data extraction For the realization of the web crawler in Python, the *Selenium* package was used in combination with Google's *ChromeDriver*. *ChromeDriver* "is an open source tool for automated testing of web apps [... and] provides capabilities for navigating to web pages, user input, JavaScript execution, and more" [2]. As, by default, a web driver can not be used from a *Python* environment, it is paired with the *Selenium Python* package which delivers the infrastructure to programmatically open a browser session and automate the web driver navigation [5].
In order to obtain structured data from the downloaded HTML files another subscript, using the *BeautifulSoup* package, is applied. "BeautifulSoup is a Python library for pulling data out of HTML and XML files" [1]. Therefore, the package is helpful in order to find and structure relevant information from the apps' HTML files. This information, including the target metadata of all applications in the accessible regions of the Google Play Store, then serves as input for the preprocessing.

4.2 Dataset (133348)

In the scope of this research only independent genres (labeled by Google) are included. This means that the sub-genres within the top-genres "Games" and "Family", are dropped and not further considered. This decision is based on the limited computational power available and to avoid categories that hold some bias, regarding its superior categories. Furthermore, other features than *app description, rating, downloads, title* and *ID* are dropped.

This results in a dataset that includes 18 different genres and 1203 different apps.

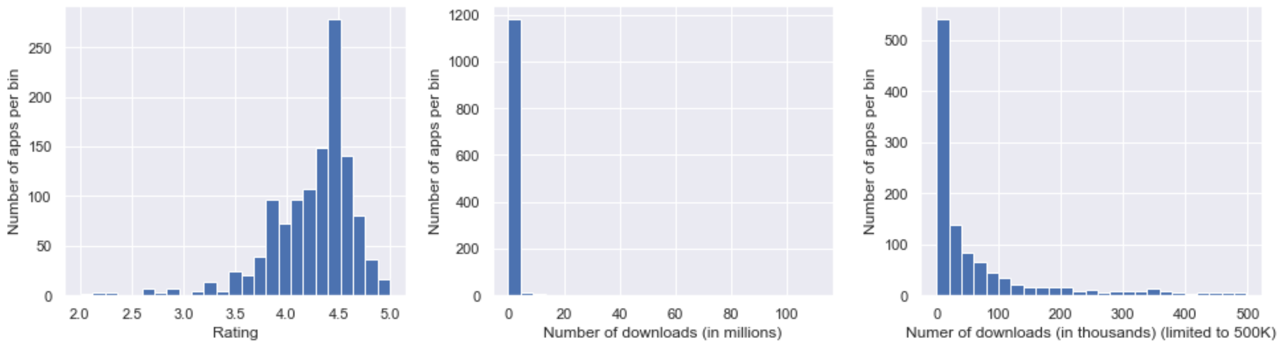


Figure 3: The distribution of the ratings (left) and their download numbers in the whole range (middle) and with a set limitation (right).

The ratings among the apps are normally distributed and slightly left skewed (see figure 3). As the median (4.3) is higher than the mean (4.24), this observation is supported by the numbers. Regarding the download numbers, most apps range within download numbers up to 200,000 with a strongly decreasing trend. However, the maximum value is 111.3 million and constitutes a strong outlier. Figure 3 (middle) and the high difference between the mean (464,020) and median (28,176) confirm this observation.

4.3 Text Preprocessing (133348)

To run analyses on textual data, it is necessary to transform the data into a standardized form. This process is known as *text preprocessing* [22, 8]. The steps this process includes will be explained in the following.

Standardization and Cleansing First, all letters of the text documents are transformed into lower-case letters. If this is not done, the words *lowercase* and *Lowercase* are recognized as different words [8]. On the downside, some information value gets lost, where the meaning derives from capitalization, such as in the example *fed* and *FED* [28]. Additionally, since commonly punctuation, numbers and special characters do not add any significant value for later analysis, they are removed [42].

Tokenization Manning et al. define a token as "an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing" [28, p.22]. This grouping can be difficult at times. Depending on the algorithm, multiwords such as *New York* could be seen as one or two tokens [22]. Another example is the word *aren't*, that could result in several different tokens ($\boxed{\textit{aren't}}$, $\boxed{\textit{arent}}$, $\boxed{\textit{are}}$ $\boxed{\textit{n't}}$ *or* $\boxed{\textit{aren}}$ $\boxed{\textit{t}}$) [28]. Some of these decisions are already taken by removing special characters beforehand.

Lemmatization and Stemming On basis of grammatical rules often different forms of a word are used in a document. By applying lemmatization, words are transformed to their base form, which is referred to as *lemma* [28]. E.g. the lemma of the words *am, are* and *is* is *be*. A simplified version of lemmatization is stemming which removes suffixes from the end of a word by predefined rules. However, only lemmatization is used in this paper, since it accounts better for morphology and therefore has proven to be more robust [7].

Stop word removal The most frequently used words, which add information-wise little or no value and solely serve grammatical purposes (such as *are, to* or *be*) are called stop words [44, 8]. They are commonly removed based on a pre-defined or manually created, so-called stop word list [22].

POS tagging Part-of-speech (POS) tagging describes the process of labelling tokens as one of eight possible POS: "noun, verb, pronoun, preposition, adverb, conjunction, participle, and article" [22, p.151] [27]. Since not all POS hold the same information value, especially for topic modelling, it can be useful to only take certain POS types into account [30].

Python offers a variety of packages for all steps of text preprocessing. In this paper the Python package *NLTK* is used. The *english* stop word list serves as basis and is expanded by the company names of the app developers and further words such as "http" are appended, since they do not add any informational value to the purpose of this work. In the LDA topic modelling only nouns were taken into account since they are specifically useful to identify topics and tend to lead to better results [30].

4.4 Text modelling

With the vast amount of information available, appropriate tools are essential to understand and summarize it in a quick manner [10]. Two approaches were selected for the purpose of the research - LDA and Doc2Vec. In the following section, it is elaborated and examined, whether these methods reveal shared characteristics of the apps based on their textual data of the descriptions.

4.4.1 LDA (133348)

One of the most popular unsupervised topic modelling approaches is the generative probabilistic model Latent Dirichlet Allocation (LDA) [11, 22]. It enables to discover efficiently underlying thematic structures within large discrete data sets such as a text corpus, by retaining important statistical relationships. It results in a probabilistic topic assignment for each document [11]. LDA bases on the simplifying assumption of *exchangeablity*, which means conditional independence and an identical distribution of the latent parameters. This means, contrary to e.g. a Word2Vec model that the context of sentences is not taken into account. However, the model can be enriched by using n-grams, but in this paper only a simple "bag-of-word" model (unigram) is used due to the scope of this paper [11].

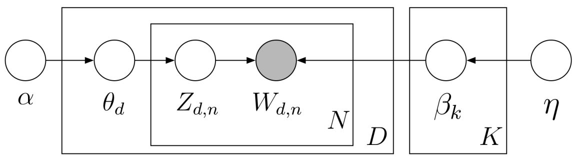


Figure 4: LDA as a graphical model [**?** , p.23].

The dependencies of the different parameters of the LDA model can be better understood with a graphical model (see figure 4), which is in "plate" notation, where every plate (shown as a box) stands for a repeated entity, called replicate. The outer level of the model, which represents the corpus, holds the proportion parameter α and the topic parameter η. α controls the topic distribution per document and the topic parameter η the word distribution per topic, both controlling a Dirichlet distribution. β_k samples the word distribution per topic k of K topics. The variable θ_d is sampled on the document level d for all D documents. On the most inner level the variable $z_{d,n}$, which assigns one topic k to each word $w_{d,n}$ in a document d, is sampled. Of all the variables in the LDA model, only the words $W_{d,n}$ (colored in grey) can be observed. All other variables are latent and therefore need to be calculated. This can be done with posterior inferences algorithms [10]. The Python package *GenSim*, which is used for LDA modelling in this paper, is based on a variational Bayes algorithm to determine the latent variables since it is able to handle massive amounts of data well [21, 4].

Generally, LDA pursues two opposed objectives. On one hand, the words of each document should be allocated only to a few topics, but on the other hand the probability for as few words as possible should be maximized for each topic. The challenge is to find the right balance between these two objectives [10].

For preprocessing, based on the research of Martin and Johnson and iterative testing, only *noun* POS are kept for topic modelling, to improve the topic's semantic coherence and as a side effect reduce the number of data points. This is possible since the LDA model works with the "bag-of-word" assumption and context is not accounted for. For the LDA model a dictionary (id2word) is created, which assigns a unique ID to each word. Subsequently, before creating a corpus, based on the ID and word occurrences, the number of words is reduced further to words with higher informative value within topics. Therefore, words which occur only in less than ten documents and words that do not occur in more than ten percent of documents are removed.

Since the expected number of searched topics is uncertain, the hyperparameters α and η are set to *auto*. To improve the model selection process the coherence measure (C_v variation) is introduced. It is used as remedy in topic modelling, to increase the interpretability of the output [41]. The score shows a conformity to human coherence judgements [34]. For 18 topics (see figure 5) the score shows a peek, which is the number of provided genres by Google, as well as 54 topics, which is proceeded with, to generate a more granular breakdown than by Google.

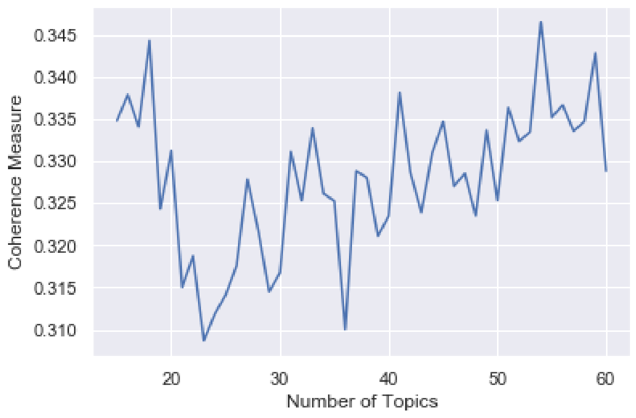


Figure 5: Coherence measure for 15 to 60 topics.

For the model evaluation, the Python package *LDAvis* is used. The topics are projected on the two-dimensional space with multidimensional scaling, to show the semantic proximity [43]. In figure 11 (see section 7) the intertopic distance plot shows that the model is able to recognize semantic differences due to the spread over the quadrants and also the range of topic importance within the corpus, captured by the size of the circles, is fine [25].

4.4.2 Word2Vec and Doc2Vec (132430)

Word2vec, compared to LDA offers the notion of context and multiple degrees of similarity (phonetic, lexical, but also semantic) between the words in the corpus [32]. It builds upon Hinton's proposal of vector representations of the words [20], combined with feed forward Neural Net Language Models (NNLM) [9] and Recurrent Neural Net Language Model (RNNLM) [33]. To reduce NNLM complexity, Mikolov et al. propose new log-linear models called Continuous Bag-of-Words (CBOW) model and Continuous Skip-gram model [32]. Their architecture can be seen in figure 6.

In CBOW, the non-linear hidden layer is removed and the projection layer of the neural network is shared for all words. The order of words does not affect the projection (hence called "bag-of-words"). But the model uses distributed representation of the context by considering n past and n future words to determine the probability of the word in the middle [32]. Skip-gram attempts to maximize the classification of words based on other

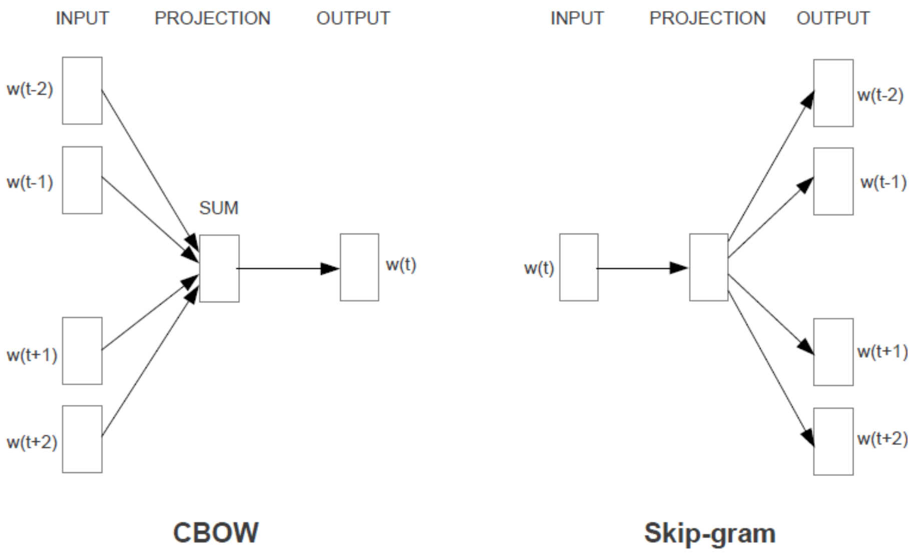


Figure 6: Architecture of Word2Vec models [32]

words in the sentence. The input is the current word and words within a certain range before and after it are predicted. The larger the context, the better the accuracy gets, but so does complexity and training time. As the words directly next to each other are more likely to be influential, the words further in the range are given less weight [31]. For the purpose of the analysis, an extension to Word2Vec, called Doc2Vec is used [3].

Doc2Vec is based on a Paragraph Vector (PV), which enables a variable length of text to be represented by vectors. It constitutes two matrices, where each paragraph is mapped into a unique vector as a column in matrix D and every word is mapped to a unique vector as a column in the matrix W [31]. The paragraph and word vectors are then concatenated to predict the next word in the context, using a softmax activation function [38], usually hierarchical, for better efficiency [31]. The PV is just like another word in terms of representation, but serves as a memory for retaining the context of the paragraph [24].

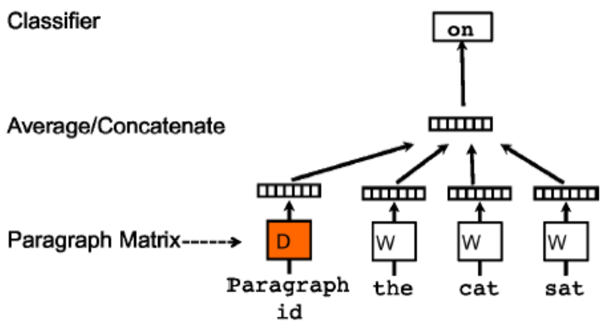


Figure 7: PV-DM architecture [31]

As in Word2Vec, two models are proposed. First, Distributed Memory Model of Paragraph Vectors (PV-DM) infers PV from unlabeled data in two stages. In the first we obtain word vectors W, softmax weights and paragraph vectors D. In the second, inference stage, the PV is obtained for previously unseen text using gradient descent with W and softmax weights unchanged from the first stage. The second model ignores the context words in the input, but lets the model predict words from the output paragraph. This method is called

Distributed Bag of Words version of Paragraph Vector (PV-DBOW) and is similar to skip-gram method from Word2Vec. It requires less memory, as it stores only the softmax weights[24].

Since the seminal Word2Vec paper, adjusting the parameters for the word embeddings has been a topic of several studies [12, 39, 23]. As the Google Play Store dataset is smaller and of a different nature than the one Mikolov et al. used in [32], extensive parameter tuning is performed. The default parameters perform well on the training dataset, but fail to generalize, implying overfitting. Thus we reduce the complexity by setting the vector size to 30. This is contrary to [12] who described the marginal benefit of embedding size tuning as not significant. On the other hand, selecting only the words with frequency of 10 or more and using a window of range 10 around the current and predicted words helps to generalize significantly better. Further, we used the default, PV-DM model and did not sub-sample any words, as the dataset is not large enough for the need to do so.

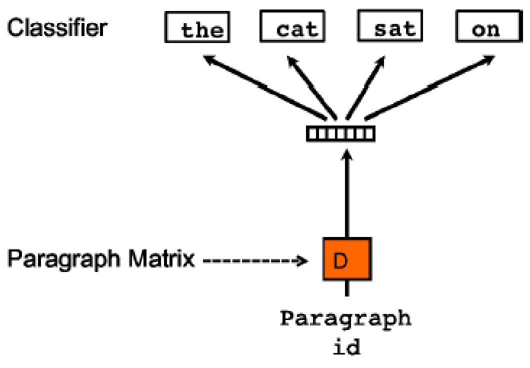


Figure 8: PV-DBOW architecture [24]

4.5 K-means Clustering (19728)

K-means is an unsupervised machine learning task that identifies similar instances and assigns them to clusters. Using k-means with the eucledian distance as distance measure will cluster documents by the amount and the meaning of the words contained. As the length of the document should not lead to different classifications, the k-means is applied on the normalized document vectors, only preserving the meaning of a document and not its length. k-means requires the user to assign the number of clusters k the algorithm must generate. The algorithm starts off by placing the k centroids (center of a cluster) randomly. It then (1) labels each instance with the centroid that is closest (2) updates the centroids by calculating the average instance-location in each cluster and moves the centroid to that average location. Step (1) and (2) are then repeated in an iterative manner, until the centroids stop moving. The computational complexity of the algorithm is generally linear with regard to the number of instances m, the number of clusters k, and the number of dimensions n.[16]

In most cases the number of k is not obvious. To find the optimal k, one can plot the inertia as a function of k. The inertia is the mean of the squared distances between each instance and its closest centroid. To only rely on the inertia when determining k is questionable, as the inertia will decrease by force as k increases. Continuously increasing k bears the risk of dividing previously reasonable clusters for no good reason. To combat this, research suggests the silhouette score as another measure to determine the optimal value of k. The silhouette score is the mean silhouette coefficient over all the instances. The silhouette coefficient is equal to $\frac{b-a}{argmax(a,b)}$, where a is the mean distance to the other instances in the same cluster, and b is the mean distance to the instances of the second closest cluster. The silhouette coefficient ranges between -1 and +1. The closer it is to +1, the closer the instance is to its own cluster and farther from other clusters. If the coefficient is closer to -1, it is possible that the the instance has been assigned to the wrong cluster.[16]

In order to determine the optimal value of clusters, the previously described approach is applied. This results in $k = 54$ for the LDA generated document vectors and $k = 33$ for the Doc2Vec generated document vectors. Whereas the inertia function indicates a proper k by abruptly reducing its negative slope, creating the typical elbow shape, the silhouette function indicates a suitable k-value with a high score value. As one can see in figure 16, both LDA related functions clearly indicate 54 as the optimal k value, which is particularly interesting, as the LDA model found exactly 54 topics in the data as well (figure 16 in appendix). In the case of the Doc2Vec model, although the silhouette score reaches its global maximum at $k = 33$, the inertia function is ambiguous.

During the analysis, it was experimented to use the DBSCAN cluster algorithm as an alternative to the k-means algorithm. The DBSCAN "defines clusters as continuous regions of high density" [16, p.287] and outliers as regions with low density of observations respectively. As reported in the code appended to this work, DBSCAN assigns 355 apps (29,5%) as outliers in the best performing state. As this setting would lead to an insufficient representation of the app market in the following analysis and was therefore decided to be excluded from the work.

4.6 Market Definition (133802)

Popularity measures can be assigned to apps. According to Zhu et al., the three main types of important popularity information are chart rankings, user ratings and user reviews, but also the download numbers of each app inside an app store [47]. Further, popularity is related to the number of users that have installed an app and currently use it [19]. Therefore, the popularity measures can be seen as indicators of the market power and revenue stream of an app. Based on this idea, apps in this work are abstracted to the two dimensional space of ratings and the number of downloads. This approach is inspired by the BCG Growth Share Matrix (compare figure 9), where businesses are abstracted to the two traditionally business critical dimensions industry growth rate and market share. According to BCG, the BCG matrix "helps companies decide how to prioritize their different businesses" within a strategic horizon [6].

In the traditional model, the industry growth rate represents the future potential and the market share represents the current market power of a business. Parallel to that, we assume the downloads to be informative about the current market power of an app and the rating of the current level of satisfaction of the users. As the literature shows that the user satisfaction level, expressed by reviews on an online platform, correlates positively with the products future sales [45]. Ratings, equally to the market growth, can be informative about future market potential.

Similar to the BCG matrix and created by the aforementioned assumptions, the two dimensional space is divided in four quadrants, low performers (low downloads, low ratings), emerging sector (low downloads, high ratings), opportunity sector (high downloads, low ratings) and top performers (high downloads, high ratings) as visualization in figure 9.

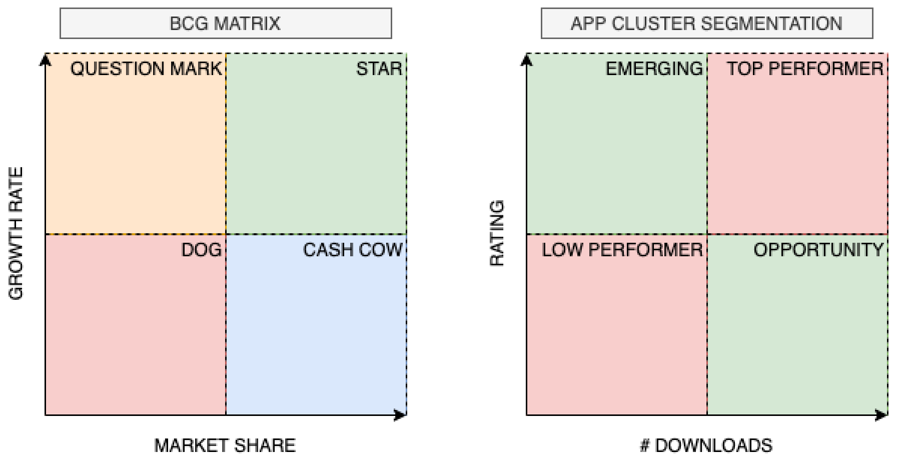


Figure 9: A visualization of the widely known BCG matrix (left) and the proposed app cluster in segmentation (right)

The difference between the BCG matrix and the proposed segmentation is the observers point of view. Whereas the BCG matrix was invented to visualize the businesses of a market incumbent, the app cluster segmentation matrix is applied by the market entrant. This is an important difference as a business, which is low on the vertical and high on the horizontal axis, is a cash cow for the first and a market entrance opportunity for the latter [6]. Whereas the position of each app cluster is determined by the average rating and log transformed average number of downloads of the apps contained by the cluster, the sector boundaries are calculated using the same measures but applied on a ground truth of the total apps contained by all clusters. The position of the clusters relative to each other and the sector boundaries can therefore provide insights about the attractiveness to create an app for a particular cluster from a developer's point of view.

5 Results (all)

In the following section, the results are elaborated on with regards to the developed propositions.

Proposition 1 (see paragraph 2) LDA produces well interpretable results based on the most occurring words across different topics. Exemplary looking at section 7 figure 12, for topic 6 the most important words *tracker, health, weight, calorie, loss* and *fitness* can be labelled as *fitness & weight loss* and for topic 12 the words *market, stock, platform* and *trade* can be labelled with the generic term *trade finance*. However, some topics do not indicate an explicit genre, for example topic 20 (section 7 figure 13) contains the words *mail, code, planet* and *payment*. Also, some thematic overlaps between topics can be noticed.

Regarding Doc2Vec, the results in terms of document similarities with previously unseen app descriptions can be seen in section 7 figure 15. The similarities in the provided documents are clearly human interpretable. For four input documents related to text messaging (figure 15 (a)), business (figure 15 (b)), fitness (figure 15 (c)) and children (figure 15 (d)), the algorithm evaluates only app descriptions related to the observed files as similar to the input text.

With respect to proposition 1, one can therefore conclude that both of the regarded algorithms are indeed capable of comprehending text and demonstrating this in the shown example cases.

Proposition 2 (see paragraph 2) The app market segmentation matrix is discussed and evaluated for both results, LDA and Doc2Vec. The matrices can be observed in Figure 10.

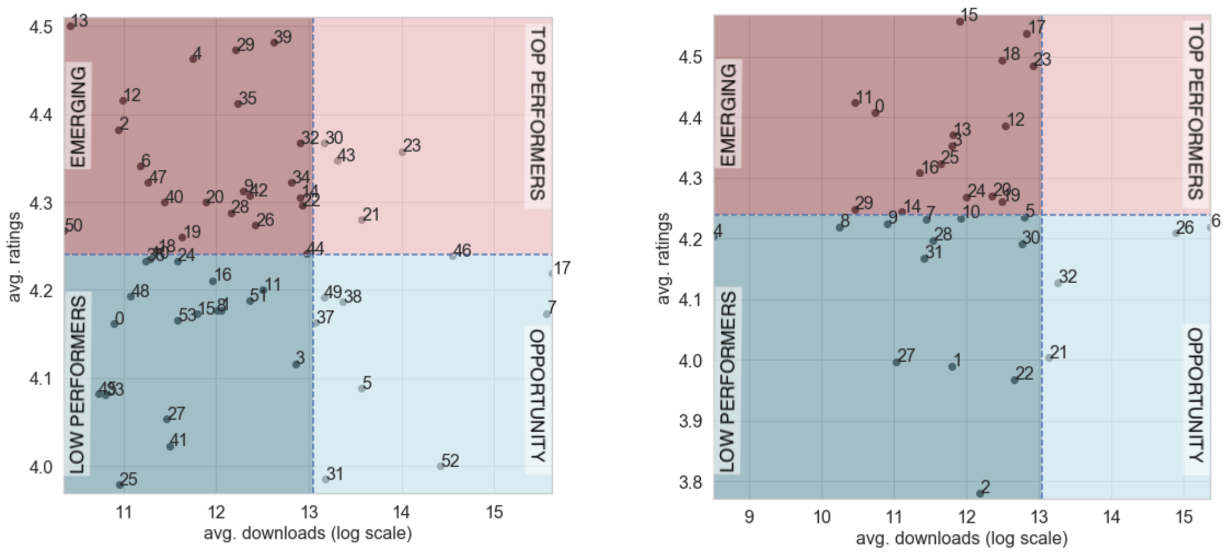


Figure 10: App cluster segmentation matrix created with k-means, based on the LDA (left) and Doc2Vec (right) output.

For LDA, the emerging segment contains a high number of clusters. For instance, cluster 4, which holds 35 apps, shows a clear focus on language studies including apps such as *Spanish Words Learn Español, English Bangla Dictionary* and *Learn French Vocabulary Free*, but it got also mixed up with apps from the fitness field, like *Fitbit Coach, Seven - 7 Minute Workout* and *Workout Trainer: fitness coach.* Cluster 13 on the other hand only holds one app and cannot be evaluated as a topic.

In the bottom right opportunity segment, the clusters are more sparse. Cluster 52 holds five apps, with all apps falling in the genre of navigation & transportation. This suggests that there is still a potential for new market entrants. Cluster 17 holds 21 apps and contains apps like *OfficeSuite Pro + PDF (Trial), Aqua Mail- Email app for Any Email* and *Hangouts* it indicates a potential in the field of communication. Lastly, cluster 31 holds 20 apps, but gives out more heterogeneous and therefore less interpretable results. Overall, the clusters are a good starting point for looking manually deeper into the markets.

Out of the 33 identified clusters by Doc2Vec, four are located in the opportunity segment. Cluster 6 holds text messaging and video apps, cluster 21 consists of entertainment apps and streaming services, 26 of browser and internet tools and 32 of transportation related apps. Compared to LDA generated topics, these topics are more clear-cut and contain higher number of apps that fall withing one category. For example cluster 6, containing 61 apps, is almost exclusively communication related. Other borderline cases show slightly more cast-away apps, but still give away a clear topic representation and outperform LDA in terms of accuracy.

The emerging segment contains the majority of the apps. Clearly free-standing clusters are related to gastronomy, fitness & well-being, educational, but also women's health apps. Similar to the opportunity sector, the clusters consist of roughly 20-40 highly relevant apps, with around 5-10% outliers.

Comparing the two models, one can observe that the results of the LDA method lead to more clusters than the results of the Doc2Vec method (54 vs. 33). LDA generated clusters are more evenly distributed in the segmentation matrix than the Doc2Vec clusters, which are more dense and closer to the mean, but show more outliers (e.g. cluster 6, 26, 4 and 2 in figure 10(right)). Nevertheless, by mapping the clusters and aligning them with the market definition part from above, some interesting findings can be inferred. For example, clusters related to communication and messaging are found on the upper right side of the opportunity sector in both models. This segment is therefore served with apps that are demanded the most, but have only average ratings. Another example is a subset of fitness apps that are expressed by one clear Word2Vec cluster (15) and several mixed clusters from LDA in the same, uppermost position, representing the emerging apps. Both of these category clusters are smaller subsets of the original Google Play genre, thus the models successfully managed to separate them from the less attractive peer sub-segments, which helps reduce the human workload during the market evaluation process.

6 Discussion and Conclusion (all)

Contribution and Discussion Overall, both modelling techniques resulted in satisfying outcomes with respect to the human understanding of the assigned similarities. However, the human judges agree that the Doc2Vec results were more consistent than the LDA results.

Also in the following step, K-Means was able to grasp the characteristics of the previously assigned topics. Especially for LDA the clustering results were convincing, since the number of 54 topics found matches with the optimal number of clusters. One contribution of the work was the developed idea of transforming the technical analysis into a business related context. Therefore, the average download numbers and ratings were taken of each cluster and put into perspective by comparing it to the overall scores. The two models produce different results and related business potential, although the cluster markets should be the same or show high similarity in the optimal case, no matter which algorithm is applied. Furthermore, the market segmentation approach of the aforementioned measures is prone to outliers, because they can distort the market attractiveness of a cluster strongly. This also applies for the matrix borders and therefore a high number of clusters is placed in the low-downloads section. From a linguistic point of few, looking at the apps within clusters, many clusters made sense regarding the genre and were able to grasp the underlying modelling of the previous step, but Doc2Vec was identified as stronger than LDA which resulted in some ambiguous clusters.

Future work The research is focused on the Google Play Store and it is not known to what extent the findings may be generalizable. Future studies could implement the market evaluation approach on different application platforms for the purpose of comparing results. Other purposes could involve data collection on user reviews following a sentiment analysis across platforms to gain further insight into category popularity and how they differ. Future studies should in general aim at developing a more sophisticated segmentation matrix to overcome the issues connected to outliers. This could be achieved by combining different clustering techniques. The experimentation of different measures is also recommended as future work (i.e. paid vs unpaid apps, smaller categories). As the final point, future researchers would benefit from investigating links between app store measures and real world business measures, such as revenue streams or startup metrics to enhance the framework in the future.

Conclusion In this work, based on a crawled Google Play Store dataset, different text analytics approaches were applied to produce a market evaluation. Although the clustering of apps and the division of the market into segments can be executed in an automated manner, the results might differ from the expected, especially using empirically obtained data. In the end, especially at an early stage, human interaction and judgement are needed to make the final evaluation to which extent the attractiveness of a cluster is given.

References

[1] Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation. URL https://beautiful-soup-4.readthedocs.io/en/latest/.

[2] ChromeDriver - WebDriver for Chrome. URL https://chromedriver.chromium.org/home.

[3] Doc2Vec Model. URL https://radimrehurek.com/gensim/models/doc2vec.html.

[4] gensim: models.ldamodel – Latent Dirichlet Allocation. URL https://radimrehurek.com/gensim/models/ldamodel.html.

[5] GitHub - SeleniumHQ/selenium: A browser automation framework and ecosystem. URL https://github.com/SeleniumHQ/selenium/.

[6] What Is the Growth Share Matrix? | BCG. URL https://www.bcg.com/about/our-history/growth-share-matrix.aspx.

[7] Eiman Al-Shammari. Lemmatzing, Stemming, and Query Expansion Method and System, 2009.

[8] Murugan Anandarajan, Chelsey Hill, and Thomas Nolan. *Practical Text Analytics*, volume 2. 2019.

[9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, Jauvinc@iro Umontreal Ca, Jaz Kandola, Thomas Hofmann, Tomaso Poggio, and John Shawe-Taylor. A Neural Probabilistic Language Model. Technical report, 2003.

[10] David M Blei. Probabilistic Topic Models, 2012. URL http://yosinski.com/mlss12/media/slides/MLSS-2012-Blei-Probabilistic-Topic-Models.pdf.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

[12] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to Recommendation: Hyperparameters matter. In *RecSys 2018 - 12th ACM Conference on Recommender Systems*, pages 352–356. Association for Computing Machinery, Inc, 9 2018.

[13] Goutam Chakraborty. Analysis of Unstructured Data : Applications of Text Analytics and Sentiment Mining. *SAS Global Forum, Washington D.C.*, (2013):15, 2014.

[14] Davis Dwight. AI Unleashes the Power of Unstructured Data, 2019. URL https://bit.ly/3bD9QkC.

[15] Bin Fu, Jialiu Lin, Lei Liy, Christos Faloutsos, Jason Hong, and Norman Sadeh. Why people hate your App - Making sense of user feedback in a mobile app store. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1288:1276–1284, 2013.

[16] Aurelien Geron. *Hands–On Machine Learning with Scikit–Learn and TensorFlow 2nd edition.* 2019. ISBN 9781492032649.

[17] Anthony R. Gray. *Studies in Economics and Business: Marketing.* Heinemann, 2000.

[18] Mark Harman, Yue Jia, and Yuanyuan Zhang. App store mining and analysis: MSR for app stores. *IEEE International Working Conference on Mining Software Repositories*, pages 108–111, 2012.

[19] Rachel Harrison, Derek Flood, and David Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1(1):1, 5 2013.

[20] G E Hinton, J L McClelland, and D E Rumelhart. Distributed Representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 77–109. MIT Press, Cambridge, MA, USA, 1986.

[21] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, pages 1–9, 2010.

[22] Daniel Jurafsky and James H. Martin. *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 2019.

[23] Jey Han Lau and Timothy Baldwin. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. Technical report, 2016.

[24] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. 5 2014.

[25] Matti Lyra. Evaluating Topic Models, 2017. URL https://www.youtube.com/watch?v=UkmIljRIG_M.

[26] Siqi Ma, Shaowei Wang, David Lo, Robert Huijie Deng, and Cong Sun. Active Semi-supervised Approach for Checking App Behavior against Its Description. *Proceedings - International Computer Software and Applications Conference*, 2(July):179–184, 2015.

[27] Christopher D. Manning, Hinrich Schütze, and Gerhard Weikurn. *Foundations of Statistical Natural Language Processing.* 1999.

[28] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*, volume 53. Cambridge University Press, Cambridge, 2008.

[29] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. Web crawling and indexes. *Introduction to Information Retrieval*, (c):405–420, 2012.

[30] Fiona Martin and Mark Johnson. More Efficient Topic Modelling Through a Noun Only Approach. *Proceedings of Australasian Language Technology Association Workshop*, pages 111–115, 2015.

[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. Technical report.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. 1 2013.

[33] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan " Honza, " ˇ Cernock´ycernock´y, and Sanjeev Khudanpur. Recurrent neural network based language model. Technical report, 2010.

[34] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, (2):262–272, 2011.

[35] Shahab Mokarizadeh, Mihhail Matskin, and Mohammad Tafiqur Rahman. Mining and analysis of apps in google play Adaptive Distributed Information Services (ADIS) View project Knowledge Environment for Interacting Robot Swarms (RoboSWARM) View project Mining and Analysis of Apps in Google Play. Technical report.

[36] Marc Najork. Web Crawler Architecture. *Encyclopedia of Database Systems*, pages 1–4, 2017.

[37] Randy Nelson. Consumer Spending In Mobile Apps Grew 17% in 2019 to Exceed $83 Billion Globally, 2020. URL https://sensortower.com/blog/app-revenue-and-downloads-2019.

[38] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. 11 2018.

[39] Bénédicte Pierrejean and Ludovic Tanguy. Towards Qualitative Word Embeddings Evaluation: Measuring Neighbors Variation. Technical report.

[40] Zhengyang Qu, Vaibhav Rastogi, Xinyi Zhang, Yan Chen, Tiantian Zhu, and Zhong Chen. AutoCog: Measuring the description-to-permission fidelity in android applications. *Proceedings of the ACM Conference on Computer and Communications Security*, 2014-Novem(November):1354–1365, 2014.

[41] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pages 399–408, 2015.

[42] Dipanjan Sarkar. *Text Analytics with Python.* Apress, Berkeley, CA, 2016.

[43] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. pages 63–70, 2015.

[44] W. John Wilbur and Karl Sirotkin. The automatic identification of stop words. *Journal of Information Science*, 18(1):45–55, 1992.

[45] Qiang Ye, Rob Law, and Bin Gu. The impact of online user reviews on hotel room sales. *International Journal of Hospitality Management*, 28(1):180–182, 2009.

[46] Lei Yin, Ruodan Lu, and Ke Rong. Automated competitor analysis using big data.

[47] Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong, Senior Member, and Enhong Chen. Popularity Modeling for Mobile Apps: A Sequential Approach. *IEEE TRANSACTIONS ON CYBERNETICS*, 45(7):1303, 2015.

7 Appendix

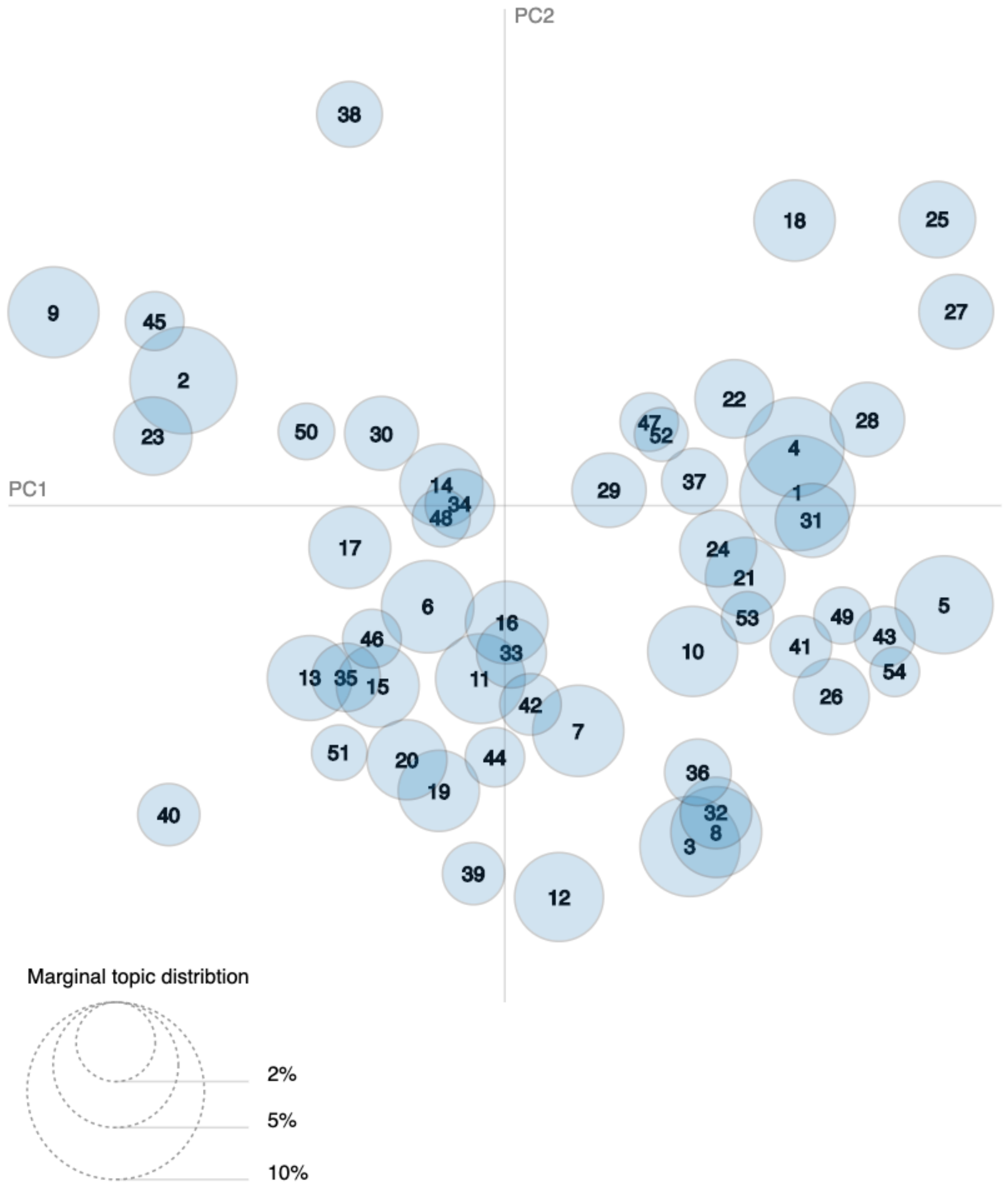


Figure 11: Intertopic Distance Map based on multidimensional scaling with *LDAviz*

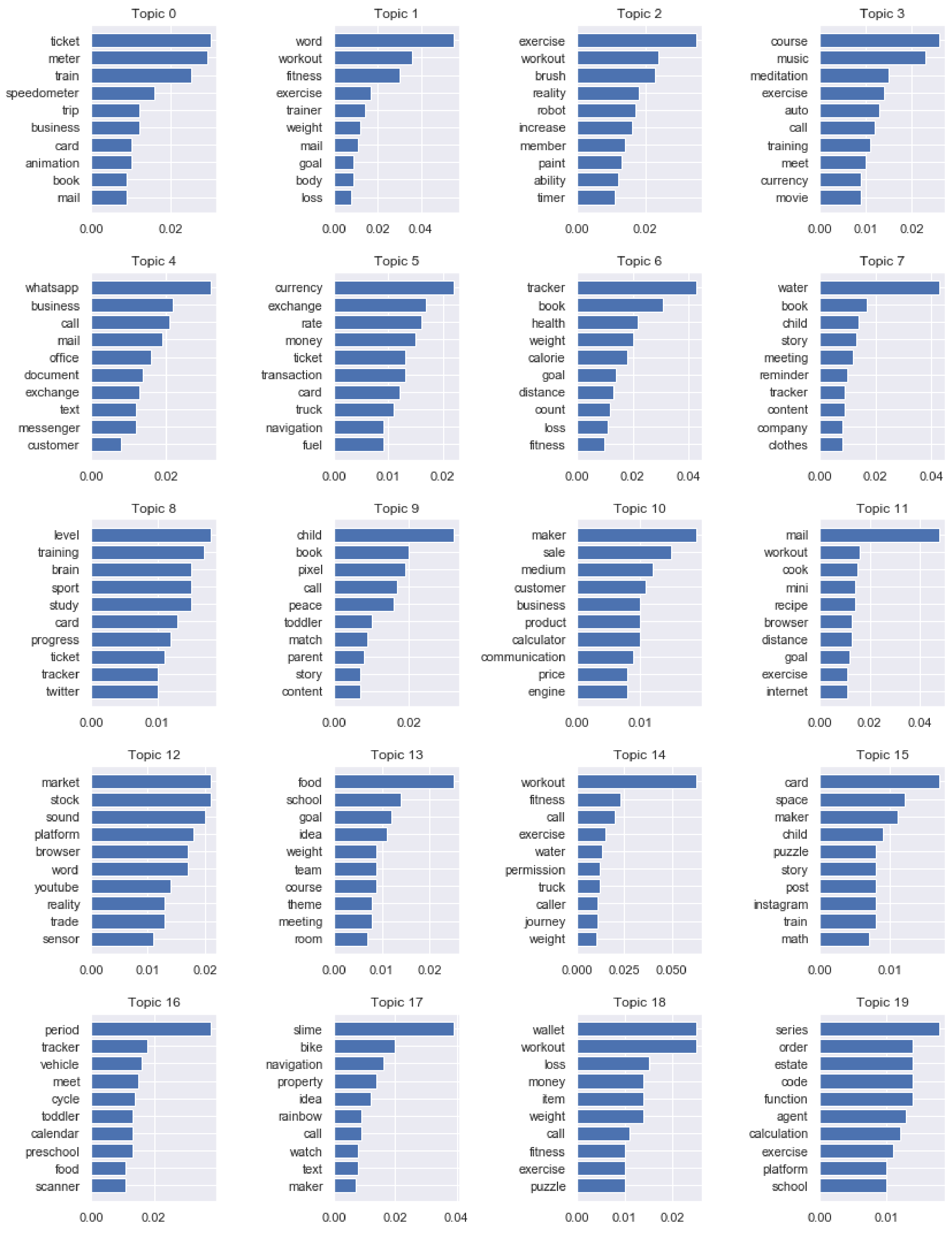


Figure 12: Ten most frequent words per topic (topic 0 to 19).

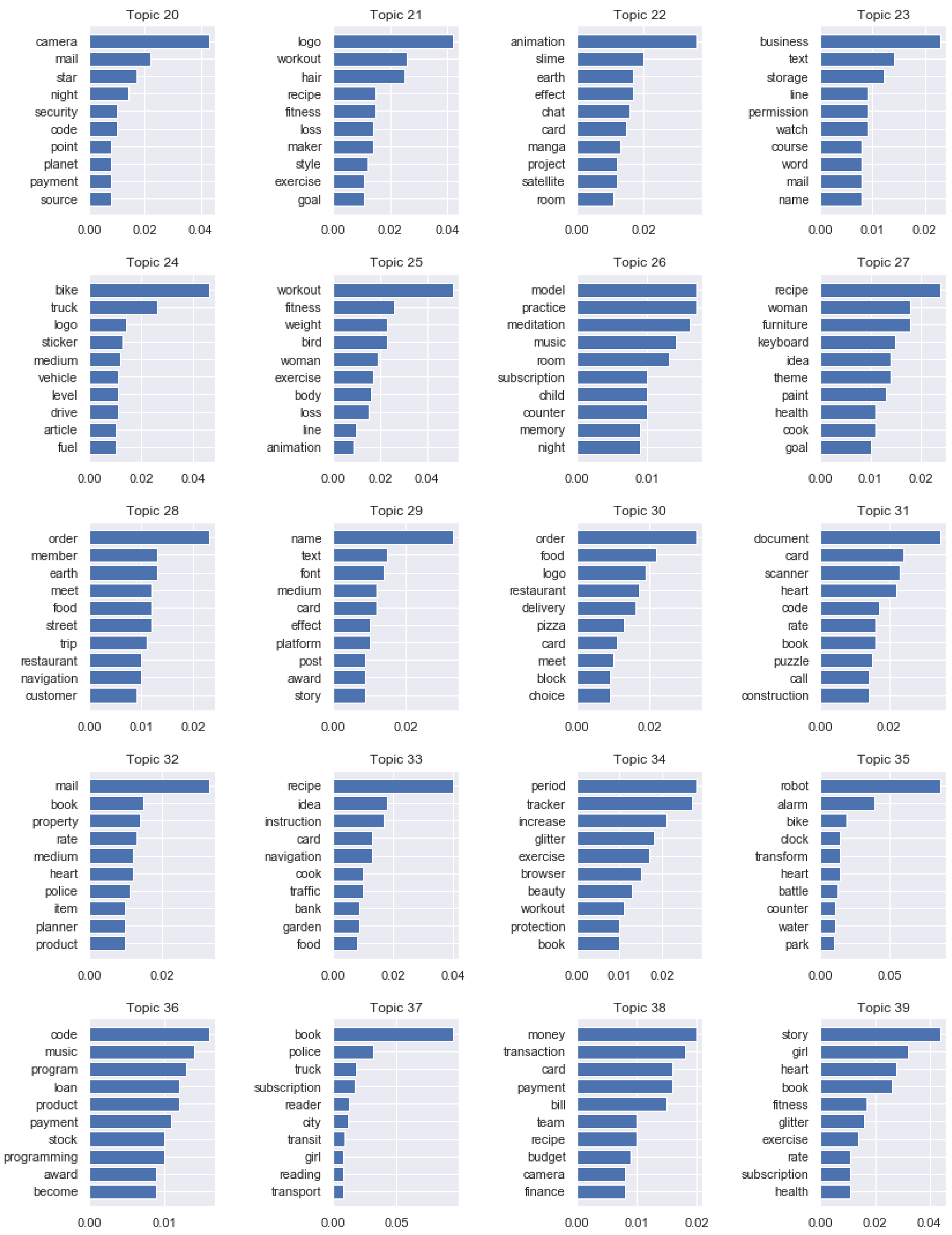


Figure 13: Ten most frequent words per topic (topic 20 to 39).

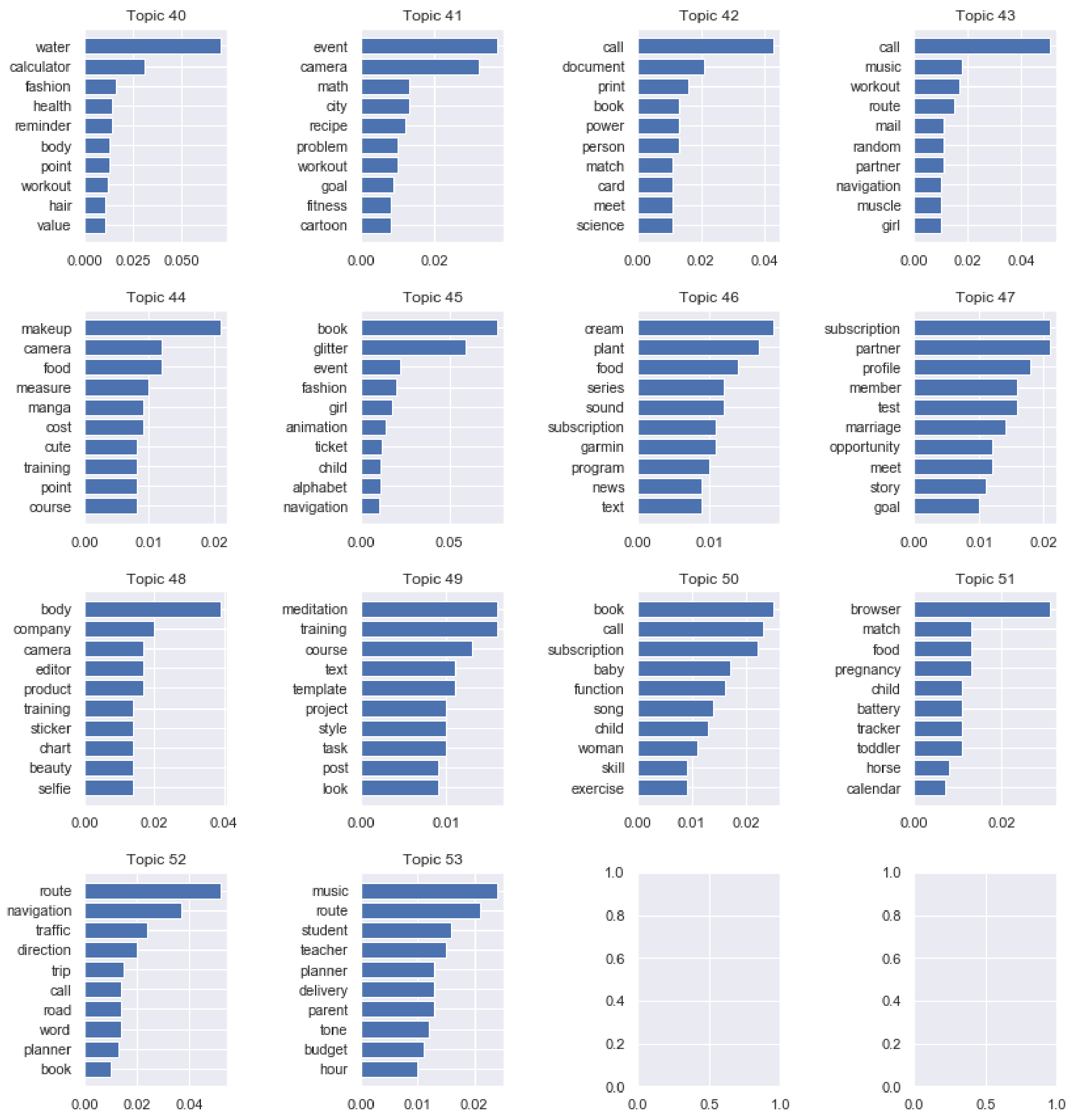
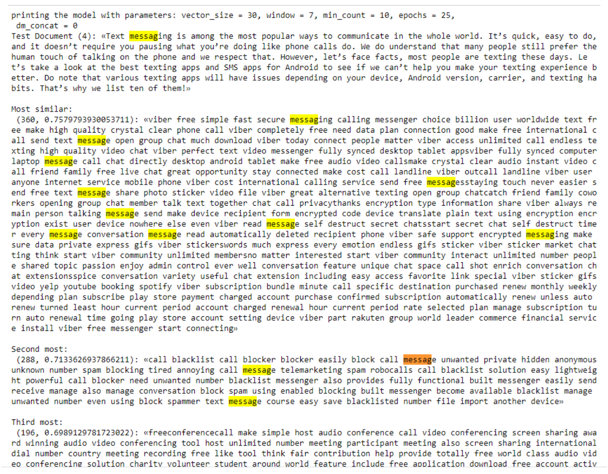


Figure 14: Ten most frequent words per topic (topic 40 to 54).

Tested on previously unseen descriptions of the apps segments taken from various sources.[1234]



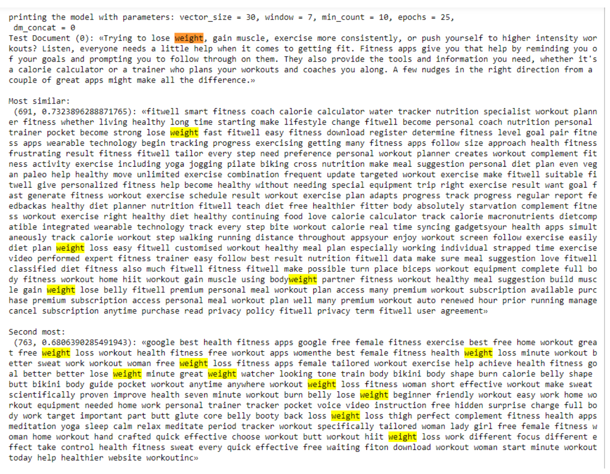
```
printing the model with parameters: vector_size = 30, window = 7, min_count = 10, epochs = 25,
 dm_concat = 0
Test Document (4): «Text messaging is among the most popular ways to communicate in the whole world. It’s quick, easy to do,
and it doesn’t require you pausing what you’re doing like phone calls do. We do understand that many people still prefer the
human touch of talking on the phone and we respect that. However, let’s face facts, most people are texting these days. Le
t’s take a look at the best texting apps and SMS apps for Android to see if we can’t help you make your texting experience b
etter. Do note that various texting apps will have issues depending on your device, Android version, carrier, and texting ha
bits. That’s why we list ten of them!»

Most similar:
 (360, 0.7579793930053711): «viber free simple fast secure messaging calling messenger choice billion user worldwide text fr
ee make high quality crystal clear phone call viber completely free need data plan connection good make free international c
all send text message open group chat much download viber today connect people matter viber access unlimited call endless te
xting high quality video chat viber perfect text video messenger fully synced desktop tablet appsviber fully synced computer
laptop message call chat directly desktop android tablet make free audio video callsmake crystal clear audio instant video c
all friend family free live chat great opportunity stay connected make cost call landline viber outcall landline viber user
anyone internet service mobile phone viber cost international calling service send free messagesstaying touch never easier s
end free text message share photo sticker video file viber great alternative texting open group chatcatch friend family cowo
rkers opening group chat member talk text together chat call privacythanks encryption type information share viber always re
main person talking message send make device recipient form encrypted code device translate plain text using encryption encr
yption exist user device nowhere else even viber read message self destruct secret chatsstart secret chat self destruct time
r every message conversation message read automatically deleted recipient phone viber safe support encrypted messaging make
sure data private express gifs viber stickerswords much express every emotion endless gifs sticker viber sticker market chat
ting think start viber community unlimited membersno matter interested start viber community interact unlimited number peopl
e shared topic passion enjoy admin control ever well conversation feature unique chat space call shot enrich conversation ch
at extensionsspice conversation variety useful chat extension including easy access favorite link special viber sticker gifs
video yelp youtube booking spotify viber subscription bundle minute call specific destination purchased renew monthly weekly
depending plan subscribe play store payment charged account purchase confirmed subscription automatically renew unless auto
renew turned least hour current period account charged renewal hour current period rate selected plan manage subscription tu
rn auto renewal time going play store account setting device viber part rakuten group world leader commerce financial servic
e install viber free messenger start connecting»

Second most:
 (288, 0.7133626937866211): «call blacklist call blocker blocker easily block call message unwanted private hidden anonymous
unknown number spam blocking tired annoying call message telemarketing spam robocalls call blacklist solution easy lightweig
ht powerful call blocker need unwanted number blacklist messenger also provides fully functional built messenger easily send
receive manage also manage conversation block spam using enabled blocking built messenger become available blacklist manage
unwanted number even using block spammer text message course easy save blacklisted number file import another device»

Third most:
 (196, 0.6989129781723022): «freeconferencecall make simple host audio conference call video conferencing screen sharing awa
rd winning audio video conferencing tool host unlimited number meeting participant meeting also screen sharing international
dial number country meeting recording free like tool think fair contribution help provide totally free world class audio vid
eo conferencing solution charity volunteer student around world feature include free application download free account activ
```

(a) Text messaging

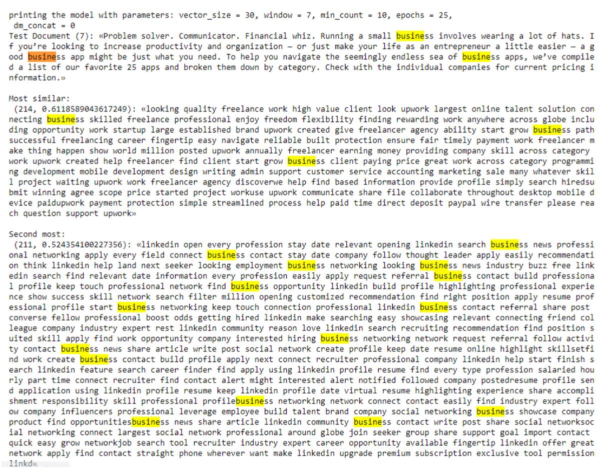


```
printing the model with parameters: vector_size = 30, window = 7, min_count = 10, epochs = 25,
 dm_concat = 0
Test Document (0): «Trying to lose weight, gain muscle, exercise more consistently, or push yourself to higher intensity wor
kouts? Listen, everyone needs a little help when it comes to getting fit. Fitness apps give you that help by reminding you o
f your goals and prompting you to follow through on them. They also provide the tools and information you need, whether it's
a calorie calculator or a trainer who plans your workouts and coaches you along. A few nudges in the right direction from a
couple of great apps might make all the difference.»

Most similar:
 (691, 0.7323896288871765): «fitwell smart fitness coach calorie calculator water tracker nutrition specialist workout plann
er fitness whether living healthy long time starting make lifestyle change fitwell become personal coach nutrition personal
trainer pocket become strong lose weight fast fitwell easy fitness download register determine fitness level goal pair fitne
ss apps wearable technology begin tracking progress exercising getting many fitness apps follow size approach health fitness
frustrating result fitness fitwell tailor every step need preference personal workout planner creates workout complement fit
ness activity exercise including yoga jogging pilate biking cross nutrition make meal suggestion personal diet plan even veg
an paleo help healthy move unlimited exercise combination frequent update targeted workout exercise make fitwell suitable fi
twell give personalized fitness help become healthy without needing special equipment trip right exercise result want goal f
ast generate fitness workout exercise schedule result workout exercise plan adapts progress track progress regular report fe
edbackas healthy diet planner nutrition fitwell teach diet free healthier fitter body absolutely starvation complement fitne
ss workout exercise right healthy diet healthy continuing food love calorie calculator track calorie macronutrients dietcomp
atible integrated wearable technology track every step bite workout calorie real time syncing gadgetsyour health apps simult
aneously track calorie workout step walking running distance throughout appsyour enjoy workout screen follow exercise easily
diet plan weight loss easy fitwell customised workout healthy meal plan especially working individual strapped time exercise
video performed expert fitness trainer easy follow best result nutrition fitwell data make sure meal suggestion love fitwell
classified diet fitness also much fitwell fitness fitwell make possible turn place biceps workout equipment complete full bo
dy fitness workout home hiit workout gain muscle using bodyweight partner fitness workout healthy meal suggestion build musc
le gain weight lose belly fitwell premium personal meal workout plan access many premium workout subscription available purc
hase premium subscription access personal meal workout plan well many premium workout auto renewed hour prior running manage
cancel subscription anytime purchase read privacy policy fitwell privacy term fitwell user agreement»

Second most:
 (763, 0.6806390285491943): «google best health fitness apps google free female fitness exercise best free home workout grea
t free weight loss workout health fitness free workout apps womenthe best female fitness health weight loss minute workout b
etter sweat work workout woman free weight loss fitness apps female tailored workout exercise help achieve health fitness go
al better better lose weight minute great weight watcher looking tone train body bikini body shape burn calorie belly shape
butt bikini body guide pocket workout anytime anywhere workout weight loss fitness woman short effective workout make sweat
scientifically proven improve health seven minute workout burn belly lose weight beginner friendly workout easy work home wo
rkout equipment needed home work personal trainer tracker pocket voice video instruction free hidden surprise charge full bo
dy work target important part butt glute core belly booty back loss weight loss thigh perfect complement fitness health apps
meditation yoga sleep calm relax meditate period tracker workout specifically tailored woman lady girl free female fitness w
oman home workout hand crafted quick effective choose workout butt workout hiit weight loss work different focus different e
ffect take control health fitness sweat every quick effective free waiting fiton download workout woman start minute workout
today help healthier website workoutinc»
```

(b) Fitness

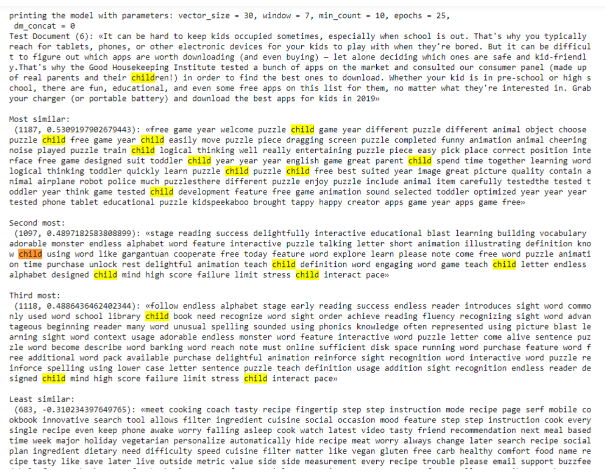


```
printing the model with parameters: vector_size = 30, window = 7, min_count = 10, epochs = 25,
 dm_concat = 0
Test Document (7): «Problem solver. Communicator. Financial whiz. Running a small business involves wearing a lot of hats. I
f you’re looking to increase productivity and organization – or just make your life as an entrepreneur a little easier – a g
ood business app might be just what you need. To help you navigate the seemingly endless sea of business apps, we’ve compile
d a list of our favorite 25 apps and broken them down by category. Check with the individual companies for current pricing i
nformation.»

Most similar:
 (214, 0.6118589043617249): «looking quality freelance work high value client look upwork largest online talent solution con
necting business skilled freelance professional enjoy freedom flexibility finding rewarding work anywhere across globe inclu
ding opportunity work startup large established brand upwork created give freelancer agency ability start grow business path
successful freelancing career fingertip easy navigate reliable built protection ensure fair timely payment work freelancer m
ake thing happen show world million posted upwork annually freelancer earning money providing company skill across category
work upwork created help freelancer find client start grow business client paying price great work across category programmi
ng development mobile development design writing admin support customer service accounting marketing sale many whatever skil
l project waiting upwork work freelancer agency discoverwe help find based information provide profile simply search hiredsu
bmit winning agree scope price started project workuse upwork communicate share file collaborate throughout desktop mobile d
evice paidupwork payment protection simple streamlined process help paid time direct deposit paypal wire transfer please rea
ch question support upwork»

Second most:
 (211, 0.524354100227356): «linkedin open every profession stay date relevant opening linkedin search business news professi
onal networking apply every field connect business contact stay date company follow thought leader apply easily recommendati
on think linkedin help land next seeker looking employment business networking looking business news industry buzz free link
edin search find relevant date information every profession easily apply request referral business contact build professiona
l profile keep touch professional network find business opportunity linkedin build profile highlighting professional experie
nce show success skill network search filter million opening customized recommendation find right position apply resume prof
essional profile start business networking keep touch connection professional linkedin business contact referral share post
converse fellow professional boost odds getting hired linkedin make searching easy showcasing relevant connecting friend col
league company industry expert rest linkedin community reason love linkedin search recruiting recommendation find position s
uited skill apply find work opportunity company interested hiring business networking network request referral follow activi
ty contact business news share article write post social network create profile keep date resume online highlight skillsetfi
nd work create business contact build profile apply next connect recruiter professional company linkedin help start finish s
earch linkedin feature search career finder find apply using linkedin profile resume find every type profession salaried hou
rly part time connect recruiter find contact alert might interested alert notified followed company postedresume profile sen
d application using linkedin profile resume keep linkedin profile date virtual resume highlighting experience share accompli
shment responsibility skill professional profilebusiness networking network connect contact easily find industry expert foll
ow company influencers professional leverage employee build talent brand company social networking business showcase company
product find opportunitiesbusiness news share article linkedin community business contact write post share social networksoc
ial networking connect largest social network professional around globe join seeker group share support goal import contact
quick easy grow networkjob search tool recruiter industry expert career opportunity available fingertip linkedin offer great
network apply find contact straight phone wherever want make linkedin upgrade premium subscription exclusive tool permission
linkd»
```

(c) Business



```
printing the model with parameters: vector_size = 30, window = 7, min_count = 10, epochs = 25,
 dm_concat = 0
Test Document (6): «It can be hard to keep kids occupied sometimes, especially when school is out. That's why you typically
reach for tablets, phones, or other electronic devices for your kids to play with when they're bored. But it can be difficul
t to figure out which apps are worth downloading (and even buying) – let alone deciding which ones are safe and kid-friendl
y.That's why the Good Housekeeping Institute tested a bunch of apps on the market and consulted our consumer panel (made up
of real parents and their children!) in order to find the best ones to download. Whether your kid is in pre-school or high s
chool, there are fun, educational, and even some free apps on this list for them, no matter what they're interested in. Grab
your charger (or portable battery) and download the best apps for kids in 2019»

Most similar:
 (1187, 0.5309197902679443): «free game year welcome puzzle child game year different puzzle different animal object choose
puzzle child free game year child easily move puzzle piece dragging screen puzzle completed funny animation animal cheering
noise played puzzle train child logical thinking well really entertaining puzzle piece easy pick place correct position inte
rface free game designed suit toddler child year year year english game great parent child spend time together learning word
logical thinking toddler quickly learn puzzle child puzzle child free best suited year image great picture quality contain a
nimal airplane robot police much puzzlesthere different puzzle enjoy puzzle include animal item carefully testedthe tested t
oddler year think game tested child development feature free game animation sound selected toddler optimized year year year
tested phone tablet educational puzzle kidspeekaboo brought tappy happy creator apps game year apps game free»

Second most:
 (1097, 0.4897182583808899): «stage reading success delightfully interactive educational blast learning building vocabulary
adorable monster endless alphabet word feature interactive puzzle talking letter short animation illustrating definition kno
w child using word like gargantuan cooperate free today feature word explore learn please note come free word puzzle animati
on time purchase unlock rest delightful animation teach child definition word engaging word game teach child letter endless
alphabet designed child mind high score failure limit stress child interact pace»

Third most:
 (1118, 0.4886436462402344): «follow endless alphabet stage early reading success endless reader introduces sight word commo
nly used word school library child book need recognize word sight order achieve reading fluency recognizing sight word advan
tageous beginning reader many word unusual spelling sounded using phonics knowledge often represented using picture blast le
arning sight word context usage adorable endless monster word feature interactive word puzzle letter come alive sentence puz
zle word become describe word barking word reach note must online sufficient disk space running word purchase feature word f
ree additional word pack available purchase delightful animation reinforce sight recognition word interactive word puzzle re
inforce spelling using lower case letter sentence puzzle teach definition usage addition sight recognition endless reader de
signed child mind high score failure limit stress child interact pace»

Least similar:
 (683, -0.310234397649765): «meet cooking coach tasty recipe fingertip step step instruction mode recipe page serf mobile co
okbook innovative search tool allows filter ingredient cuisine social occasion mood feature step step instruction cook every
single recipe even keep phone awake worry falling asleep cook watch latest video tasty friend recommendation next meal based
time week major holiday vegetarian personalize automatically hide recipe meat worry always change later search recipe social
plan ingredient dietary need difficulty speed cuisine filter matter like vegan gluten free carb healthy comfort food name re
cipe tasty like save later live outside metric value side side measurement every recipe trouble please email support buzzfee
```

(d) Children

Figure 15: Similarities with previously unseen text

[1]https://www.androidauthority.com/best-texting-apps-for-android-653970/, accessed on 15/06/2020
[2]https://uk.pcmag.com/mobile-apps/42450/the-best-fitness-apps-for-2020, accessed on 15/06/2020
[3]https://www.nerdwallet.com/blog/small-business/20-apps-small-business-owners/, accessed on 15/06/2020
[4]https://www.goodhousekeeping.com/electronics/g28212386/best-apps-for-kids/, accessed on 15/06/2020

```
In [44]:  kmean_test_n_clusters(doc2top_norm, 100) # 54
```

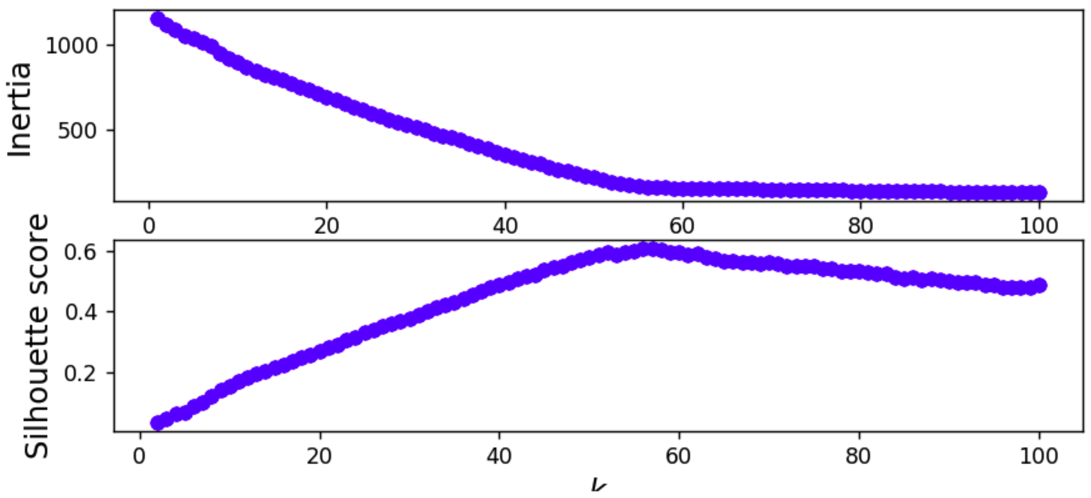


Figure 16: Inertia function and Silhouette function applied on the LDA model with a maximum k of 100.