Measuring Product Similarity A Multidimensional Approach

Department of Computer Science

Ariel University

Shoval Tayro , Yarden Cohen and Uriya Havshush

Abstract

Across many industries, recommender systems are playing an increasingly important role in enhancing user experience and boosting sales and services. On Netflix, for example, 80% of movies watched are now the result of recommend-er systems.

In case of flight destination Recommendation systems, the data are more challenging to analyze due to extra sparseness, dispersed user history actions, fast change of user interest and lack of direct or indirect feedbacks. We found that most of the Recommendation system in this case build based on text similarity.

In this paper, we propose several product-offering methods in the context of tourism, where products are flights destinations, by researching the similarity of flights destinations based on images and text trying to examine which method more represents the opinions of the people.

Basically, we use text, images, history search as input, and tries to understand the information about products from the text, the images, the search history of people and the combination of these three – we use a neural network model to learn word associations from the data we collect to embed the data. Then use another neural network to model the similarity score between pair cities, which will be used for selecting the closest product in our database. We use Cosine similarity to calculate the similarity score for training data. We collect product information data (including image, class label etc.) from Google and Lonely Planet to learn these models. Specifically, our dataset contains information about 24 products. In our research we try to achieve a classification accuracy of flight destination. Finally, we hope to offer fast and accurate online shopping support.

Contents

1 Introduction **2**

2 Related Work **4**

3 Research Methods **6**

3.1. Image-based Similarity 6

3.1.1. Classification 6

3.1.2. Recommendation 7

3.2. Text-based Similarity 8

3.2.1. Module 1: TF-IDF 8

3.2.2. Module2: bag-of-words 9

3.2.3. Module 3: Classification using Machine Learning 9

3.3. Product-network-based Similarity 10

3.4. Combination of methods 11

3.5. Comparing against the Truth Using Amazon Mechanical Turks 11

4 Results **13**

4.1. Image-based Similarity Results 13

4.2. Text-based Similarity Results 15

4.3. Product-Network-based Similarity Results 16

4.4. Combination Of Methods Results 17

4.5. Comparing against the truth using AMT Results 18

5 Conclusion **21**

6 References **22**

CHAPTER 1

INTRODUCTION

Ṭhe online retail ecosystem is fast evolving, and online shopping is unavoidable growing around the world. Digital analytics firm eMarketer shows that online retail sales continue double and account for more than 12% of global sales by 2019. As reported in the result of the Nielsen Global Connected Commerce Survey (2015), 63% of respondents who shopped or purchased the travel products or services category, for example, in the past six months say they looked up the product online.

However, the explosive growth in the amount of available digital information has created the challenge of information overload for online shoppers, which inhibits timely access to items of interest on the Internet. This has increased the demand for recommendation systems and intelligent comparison tools. Though almost every e-commerce company nowadays has its own recommendation system that can be used to provide all sorts of suggestions, they are mostly collaborative filtering-based and usually rely on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The system generates recommendations using only information about rating profiles for different users or items. One of the advantages of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. With the rapid development of neural network these recent years, we can now change the traditional search paradigms from methods that are based solely on the past interactions recorded between users and items to produce new recommendations to visual discovery. with the advance of technology, we can develop new methods using deep neural networks in order to find similarity between products without early classification of products made by content experts. In our days A snapshot of a product tells a detailed story of its appearance, usage, brand and so on. While a few pioneering works about image-based search have been applied, the application of image matching using artificial intelligence in the online shopping field remains largely unexplored. Based on this idea, in this study we test and compare new methods we developed for comparing products. these methods take advantage of deep-neural-networks and offer a model for finding similar products that combines these methods and shows that its performance is better than that of each method individually.

In our case the recommendation system we want de develop is for flight, where the products are flights destination, represented in three aspects of the product: text describing the flight destination, pics of the destination and product-network.

The input to our algorithm is an images/text description of any destination we use for the research. For images we then use image2vec model- a Deep Neural Network (DNN) model that extract vector embeddings from all images. we use the input vector of the last fully connected layer as a feature vector to feed in a similarity calculation DNN model to find the closest products in our database. For text we use 'Google's universal sentence encoder'- an Encoder that encodes text into high dimensional vectors that can be used for text classification, semantic similarity, we use this vectors in several similarity calculation between the products.
More concretely, the two functionalities that we want to achieve in developing the modules are:

1. given the features of the pics and text that this product belongs to, calculate similarity scores and find the most similar products in our database. Ideally, people looking for a flight should be recommended similar destination.

2. combine the methods mentioned above and explore if its performance is better than each method individually.

Chapter 2

Related Work

As we explained above, most of the recommendation systems were collaborative filtering based and usually rely on the assumption that people. With the advance of technology, in our work we develop modules using Machine learning methods in order to find similarity between products and can use these models in a recommendation system or other tools.

In paper [14] there is a recommendation system photo & text based, but recommendations are produced by comparing the query against the representative tags or representative images under the premise of "if you like that place, you may also like these places", we didn't use the assumption of collaborative filtering and use machine learning methods for recommendations. in [3] and [5] build a recommender system based on pics and the evaluation of people using collaborative filtering in our project we didn't use collaborative filtering and even combine methods(multi-modal-similarity). In the field of image recommendation Paper [4] presented AlexNet model that can classify images into 1000 different categories. In addition, paper [6] presented VGG neural network that classify images in ImageNet Challenge 2014. In our first part of project, we use both models to classify the categories of the products.

However, both papers did not present a method for image recommendation. Although there are papers that studies image similarity such as [9] and [8], most of them are based on category similarity, i.e., products are regarded as similar if they are in the same category. However, products that come from the same category can still vary a lot. The idea could be also found in [2] and [7].

In the field of text recommendation, before we recommend, we need to answer What is the measurement of similarity. The most nature answer is either cosine similarity or L2 norm similarity. Another way to measure the similarity is by introducing semantic Information. The paper [1] indicates that visual similarity and semantic similarity are correlated. Thus, we try introducing a new model to calculate similarities between images based on semantic information and text description. Paper [15] has used the Word2vec algorithm to form location-based data into the vector space form, then recommendations are made based on similarities Paper [11] and [10] share some of same idea as we do here. Another recommendation method that uses techniques from natural language processing is Socio-Historical method proposed in [16]. It is one of the state-of-the-art methods for venue recommendation on LBSNs. Observing the similarities in text mining and social network datasets, it employs language models approach from natural language processing to make venue recommendations. It models either users' historical preferences or their social interactions or both together, while in our models we use data that

does not need any of them.
In the field of product-network recommendations, they are mostly used for social network and rely on the network structure [17] & [18], in our case we combine few models together and examine the product-network compared to AMT results.

Chapter 3

Research Methods

The large amount of user information available is exploited by the travel recommender systems to provide suggestions to the user in the effective manner. The tourism/travel recommender system employs many techniques to generate good recommendations to the user. This section depicts the applications of the recommender systems in the field of flight destinations Tourism.

In order to examine which method more represents the opinions of the people we try several methods: image-based similarity - try to find similarity between flight destination by the similarity of the city. Text-based similarity - try to find similarity between flight destination by the similarity of text describing the city. Product-network similarity - try to find similarity between flight destination based on 'history search' of users.

§3.1. Image-based Similarity

we used pre-trained neural network models to train and extract vector embeddings from all images.

3.1.1 *Classification*

In this step, we would like to classify an input image into one of the 20 categories. We use AlexNet model.

AlexNet: a deep convolutional neural network classification model proposed by [4]. Two important contributions of the AlexNet are popularizing usage of the non-saturating rectified linear unit activation function, ReLU(x), max(0, x), and introducing a normalization layer after the ReLU activation. Empirical results show that the normalization layer improves the generalization ability of the network. AlexNet is an incredibly powerful model capable of achieving high accuracies on very challenging datasets. As we can see for these reasons, we AlexNet model is widely common for developing recommendation systems, for example [4] [12] [13] use this model for building a recommendation system using pics. As we can see, AlexNet model first contains 2 convolutional layers with max pooling and batch normalization; then there are 3 convolutional layers with separated feature; one max pooling before three fully connected layers. The original model was trained to classify images in the ImageNet LSVRC-2010 content, where there were 1000 categories. Since our problem only contains 20 categories, we change the last fully connected layer to 4096×20.

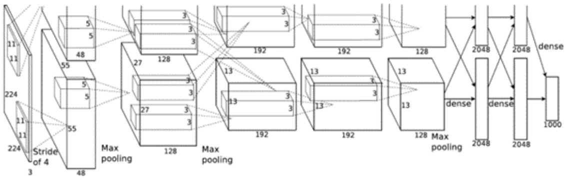


Figure 3.1: AlexNet model figure.

3.1.2 *Recommendation*

For the recommendation step, we use the last fully connected layer in our classification model as feature vectors of images. For any images in the dataset, there will be one corresponding feature vector. And this feature vector will be the input for our recommendation model. The workflow of this step is shown in the following bullets.

- Feature extraction: the classification model is used to identify which category the target image belongs to. Then we extract the input from last fully connected layer of classification model as features.

- Input of the model: the feature vector of the target image extracted in the above.

- Similarity calculation: using different measures to calculate similarity scores between feature vector of target image and feature vectors of all images in the target category to measure similarity between image pairs. We have tried cosine distance and neural network models to compute the similarity scores. The cosine distance score is defined as:

  $$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} \tag{3.1}$$

  $whereb, e \in R$ are the two corresponding feature vectors, and l = 4096 is the length of feature vectors.

  The larger the score is, the more similar the two images are. Output: top K images (products) that are most similar to the target product/image.

To calculate the similarity values based on the images of the destinations, we implemented three different methods:

Method 1:

Calculate for each matrix between two destinations the sum of the similarity values. This sum will be the similarity between the destinations. The value of the similarity between destination X and destination Y is a value obtained by the summing up of the cosine similarity between the i'th image from X and the j'th image from Y. The higher amount, the greater the similarity between the two destinations.

Method 2(Extension of Method 1):

In this method we calculate the mean and the standard deviation of similarity values across all the destinations and all the images. The value of the similarity between two destinations is made by summing up the number of cells in a matrix whose value is greater than at least the standard deviation + mean.

The value obtained is the similarity between the destinations.
Method 3:
We calculated the similarity between the destinations in percentages as follows: For each image in destination X we need to find which image from the all other destination is most similar to. Once we have found the most similar image from destination Y then the similarity between destination X and destination Y gets 5
Example of performing the calculation- We took the first picture of Paris and looked which picture most resembled it from all the other destinations pictures. We found that the image most similar to it is image number 4 of Rome. So, we added to the similarity value between Paris and Rome 5% and so on for all the pictures of all the destinations.
We have finally reached a point where each target has the percentage where it is similar to any other destinations.

§3.2. Text-based Similarity

For getting the text similarity results we develop several tools and examine them.

3.2.1 *Module 1: TF-IDF*

TF-IDF Algorithm TF-IDF is an information retrieval(IR) algorithm based on the occurrence of keywords in the whole dataset as well as particular documents. TF-IDF algorithm represents the importance of the word, Term Frequency (TF) and Inverse Document Frequency (IDF) are associated with the word importance. TF represents the number of times of a word appearing in a document. The importance of word ti in a document can be expressed as:

$$(1)TF(word) = \frac{Count(word)}{\sum_{i=1}^{n} Count(word_i)} \tag{3.2}$$

In formula (1), Count(word) presents the number of occurrences of the word in the document. The denominator is the sum of the number of the occurrences about all the words in the documents. IDF is a measure of the word ability to distinguish between categories. IDF of a word can be obtained by the total number of documents that contain the word divided by the total number of documents after the quotient logarithmic.

$$(2)IDF(word) = \mathbf{log}(\frac{Count(docs)}{Count(word, docs)} + 0.01) \tag{3.3}$$

In formula (2), Count(docs) is the total number of documents. The denominator is the total number of documents that contain the word. Calculated inverse document frequency indicates that the number of documents that contain a word fewer, Count(word ,docs) smaller, thus indicating that the word has a very good class discriminative.

$$(3)Weight(word) = TF(word) * IDF(word) \tag{3.4}$$

Formula (1)'s result is multiplied by the result of Formula (2) to obtain the result of Formula (3), which represents the weight of words.

Outcome of this module generates the product sum for every document which will help to evaluate the similarity between flight destinations.

3.2.2 *Module2: bag-of-words*

Bag-of-words is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multi-set) of its words, disregarding grammar and even word order but keeping multiplicity.
Bag-of-words model is mainly used as a tool of feature generation. After transforming the text into a "bag of words", we can calculate various measures to characterize the text. The most common type of characteristics, or features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text.

3.2.3 *Module 3: Classification using Machine Learning*

Machine learning is used to first learn the concept and then apply intelligence to take a decision. This module will help to learn about the thought behind categorization of flight destination to categorize unlabeled flights into categories. Here, the Naïve Bayes classification algorithm has been used to classify the data into multiple categories. Initially, a training step will be used to provide learning to data then work will be classified according to learn thought and during the testing module.
For embed and learn the similarity of the text describing the flight destination we used 2 main methods:

- Doc2vec- use a neural network model to learn word associations from a large corpus of text using word2vec algorithm.
- 'Google's universal sentence encoder'- use a model that already build and trained.

word2vec- The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.

word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space
The result is a set of word-vectors where vectors close together in vector space have similar meanings based on context, and word-vectors distant to each other have differing meanings. For example, strong and powerful would be close together and strong and Paris would be relatively far. With the Word2Vec model, we can calculate the vectors for each word in a document. But what if we want to calculate a vector for the entire document, we use Doc2Vec algorithm, which usually outperforms such simple-averaging of

Word2Vec vectors.

We use Paragraph Vector - Distributed Memory (PV-DM). PV-DM is analogous to Word2Vec CBOW. The doc-vectors are obtained by training a neural network on the synthetic task of predicting a center word based an average of both context word-vectors and the full document's doc-vector. the sketch above is a small extension to the CBOW (Continuous bag of words) model. But instead of using just words to predict the next word, we also added another feature vector, which is document unique.

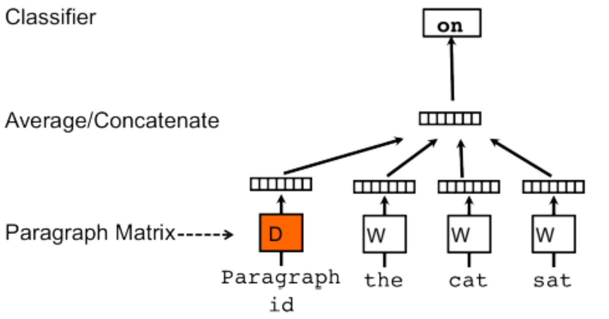


Figure 3.2: Architecture of PV-DM (Mikolov et al., 2014).

So, when training the word vectors W, the document vector D is trained as well, and in the end of training, it holds a numeric representation of the document. Then we use these embedded vectors to calculate similarity.

'Google's universal sentence encoder' - 'Google's universal sentence encoder' is a trained and optimized model for encoding sentences, phrases, or short paragraphs. It is trained on a variety of data sources and a variety of tasks with the aim of dynamically accommodating a wide variety of natural language understanding tasks.

When fed with variable-length English text, these models output a fixed dimensional embedding representation of the input strings. They take lowercase PTB tokenized string as input and output sentence embedding as a 512-dimensional vector.

We use the Deep Averaging Network (DAN) model input embeddings for words and bi-grams are averaged and fed to a feed-forward DNN (Deep Neural Network) resulting in sentence embeddings.

Google sentence encoder give us the best results based on text, so we choose this method to forward comparing at the end of the research.

§3.3. Product-network-based Similarity

Product Network Analysis is derived from the Brand Network Analysis method originally invented at the University of Cologne. Its purpose is to visualize consumers' associations regarding you or your competitors' products by means of a network graph. From these graphs one may easily read and compare the characteristics of consumer perception:

We started our designing with few assumptions :

- Similar products should be connected with each other
- Each relationship is enriched with additional information such as confidence and its source

- Relationships are saved permanently, not dynamically calculated
- Products and their relationships can be easily updated

We prefer to generate a network of products based only on transactions where each destination is linked to others because they appear in the search history from the same people, this kind of network is named products network.

The building process of directed weighted product network is divided in two equally important phases. The first one is to build a set of history search information from a lot of people. Once we have this information, we start to build our product network where each destination is linked to the next destination that appear after him in the search. These networks can be represented by an adjacency matrix showing the weight between each pair of destination. The weight is the number of times destination X appeared after destination Y in the history search.

§3.4. Combination of methods

Our hypothesis is that using a combination of the three methods will constitute the method that is more representative of people's opinions for the similarity between the different destinations.

The most common approach to get a combination of methods is to aggregate the results by using a mean (or a weighted mean). The goal of this approach is to find a compromise between the results of the different methods.

we implemented this approach in order to get the combination of all three methods that we have implemented so far - the text, the images and the search history. we calculated the mean of the result matrices of the text, the images according to the first method and the searches history.

§3.5. Comparing against the Truth Using Amazon Mechanical Turks

For evaluating the performance of each of the proposed methods for measuring the similarity between products explained above we use Amazon Mechanical Turks.

Amazon Mechanical Turks is a crowdsourcing marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed workforce who can perform these tasks virtually.

We use AMT for collecting similarity information on destinations from ten people via questionnaires. Each questionnaire was composed of 12 questions, one question for each destination name. In each question the people had to rate on a scale of 1 to 7 (where 7 is the highest score) what the level of similarity of this destination with each of the other 11 destinations in the list is.

After collecting the results, we create a heat map and use it to compare the heatmaps of the results of the methods of the images, text, search history and the combination of all three to the results we received from AMT which for us best reflect the true opinions of humans. Decide which method of imagination is most representative of human outcomes.

We compared the AMT results using 4 different methods:

1. In the first method to compare the results of the different similarity methods to the AMT results we performed a distance calculation between each matrix representing the similarity results according to text / images / search history or a combination of these methods. We calculated and sum the differences between all the corresponding cells in the matrices | Xi,j - Yi,j| Where X is the result matrix of each of the imaginary methods we implemented, and Y is the AMT result matrix.

2. In the second method, to compare the results of the different similarity methods to the AMT results, we performed the following calculation - For each matrix that represents the results of a particular similarity method we have defined for it a threshold which represents the median of that matrix. Each cell in the matrix which is equal to or greater than threshold we became 1 and the rest of the cells we became 0. We then summed all the cells in the matrix and this sum constituted the score for comparison against the AMT results matrix. The result matrix of the method that received the highest amount is the matrix that most represents people's opinions for similarity between destinations.

3. In the third method to compare the results of the different similarity methods to the AMT results we performed the following calculation- For all matrices we have defined a uniform threshold - the median of the result matrix according to AMT. Each cell in the matrix which is equal to or greater than the threshold we became 1 and the other cells we became 0. We then summed all the cells in the matrix and this sum constituted the score for comparison against the AMT results matrix. The result matrix of the method that received the highest amount is the matrix that most represents people's opinions for similarity between destinations.

4. 4. In the fourth method, to compare the results of the different similarity methods with the AMT results, we performed the following calculation: For all the matrices we defined a uniform threshold - the average of the result matrix according to AMT. Each cell in the matrix which is equal to or greater than the threshold we became 1 and the other cells we became 0. We then summed all the cells in the matrix and this sum constituted the score for comparison against the AMT results matrix. The result matrix of the method that received the highest amount is the matrix that most represents people's opinions for similarity between goals.

Chapter 4

Results

Here are the heatmaps that represent the results of each method mentioned above, text, three methods for the images, history search and the combination of all three methods. each heatmap was normalized to scale of 0-1 by min-max scale:

$$X_{norm} = [(X - min)/(max - min)] * (max - min) + min. \tag{4.1}$$

§4.1. Image-based Similarity Results

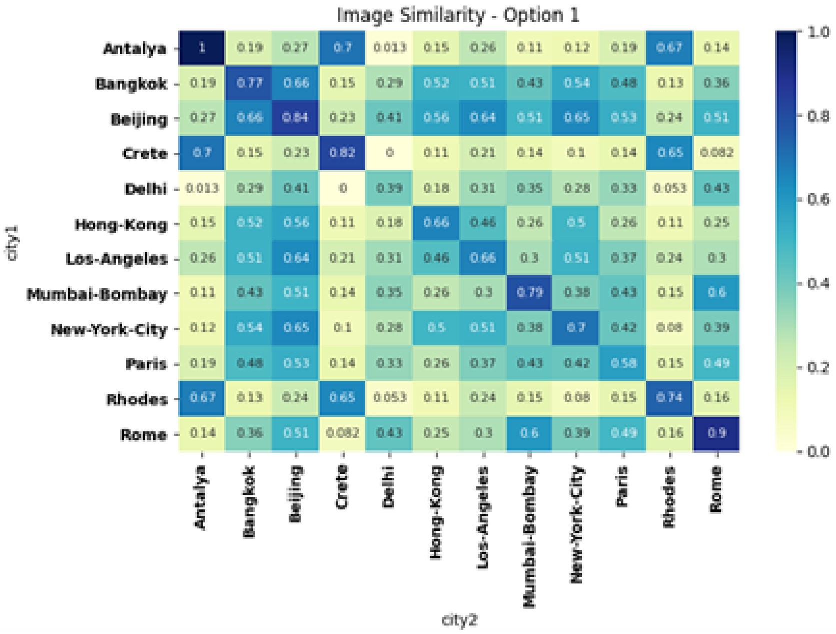


Figure 4.1: image similarity option 1

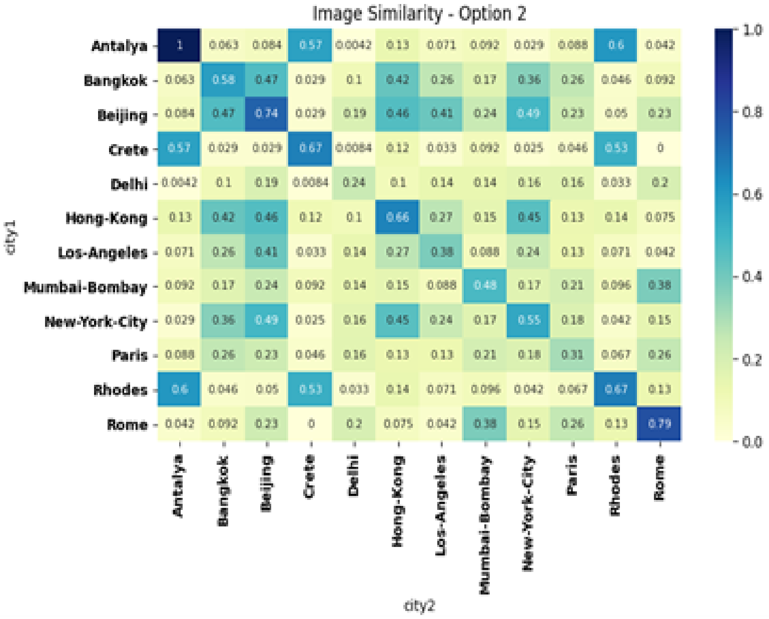


Figure 4.2: image similarity option 2

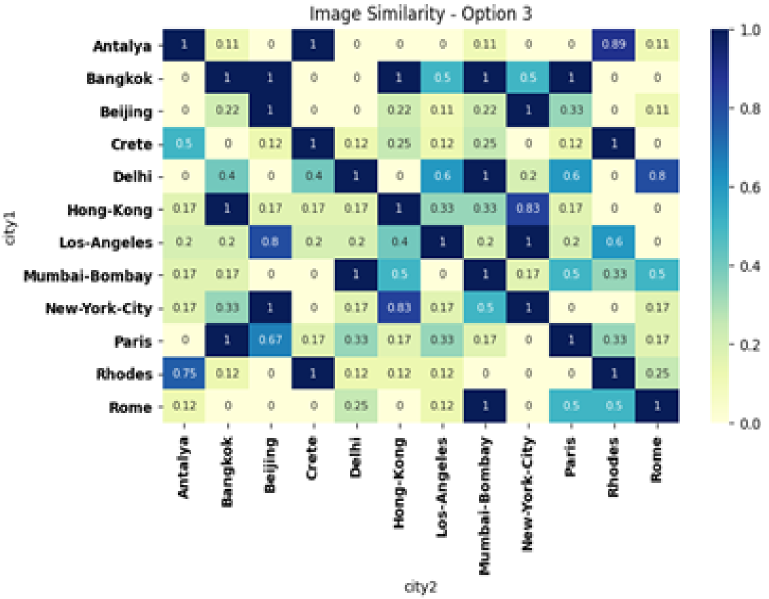


Figure 4.3: image similarity option 3

After we implement the three methods for measure the images similarity and getting their results, we choose to calculate the combination of the methods with the first method of the images(Figure 4.1) since we got a heatmap which apparently is closest to the heatmap of the AMT results and we found him more represents the opinion of the people.

§4.2. Text-based Similarity Results

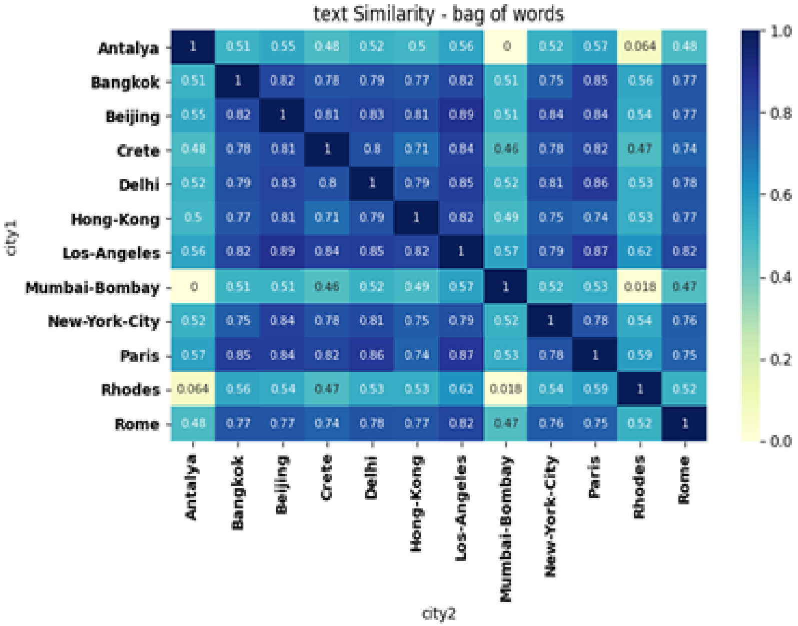


Figure 4.4: bag-of-words results

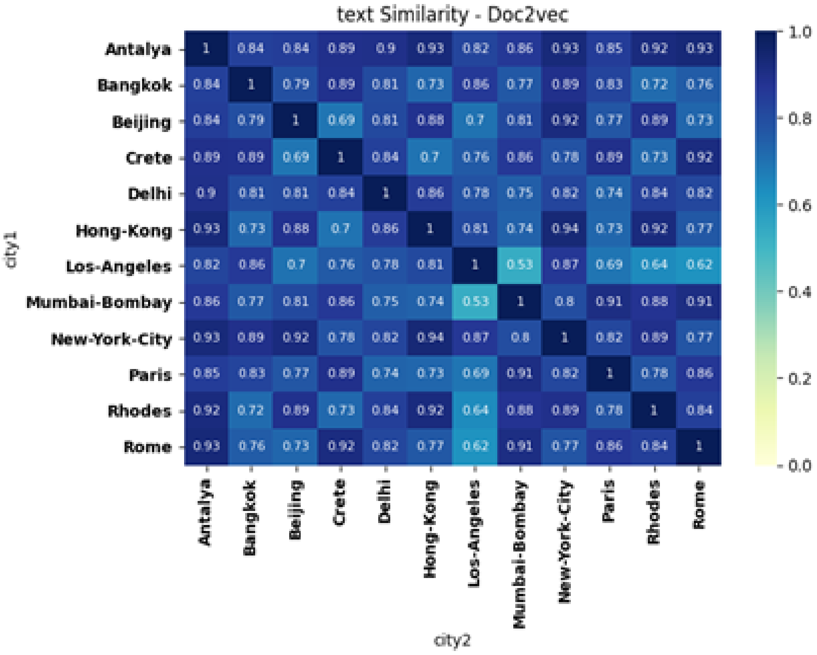


Figure 4.5: doc2vec results.

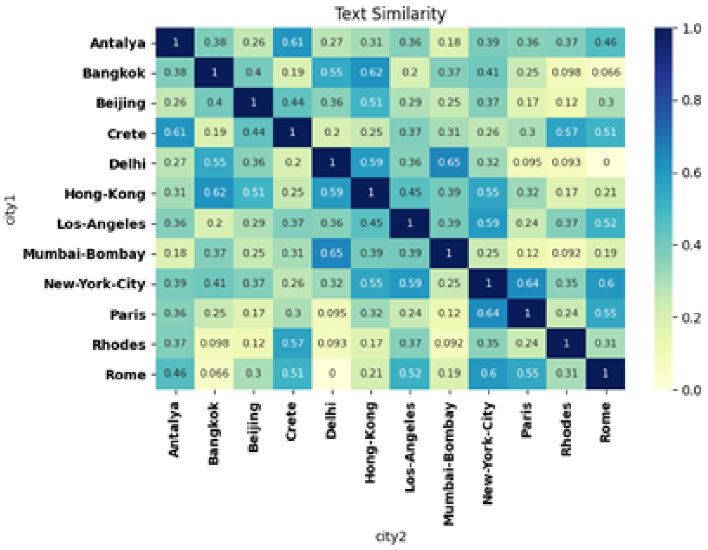


Figure 4.6: 'google-encoder' results

As we can see the results of the text similarity according to 'bag-of-words'(Figure 4.4) and Doc2vec(Figure 4.4) modules gave us 'good' results in aspect of numbers, but they are not representative enough since for a text the semantics of the sentence should be given importance as well and not only the words themselves.Another reason we choose 'google-encoder' over word2vec is that 'google-encoder' is a trained and optimized model for encoding sentences. Due to these reasons we choose to use 'google-encoder' results(Figure 4.6).

§4.3. Product-Network-based Similarity Results

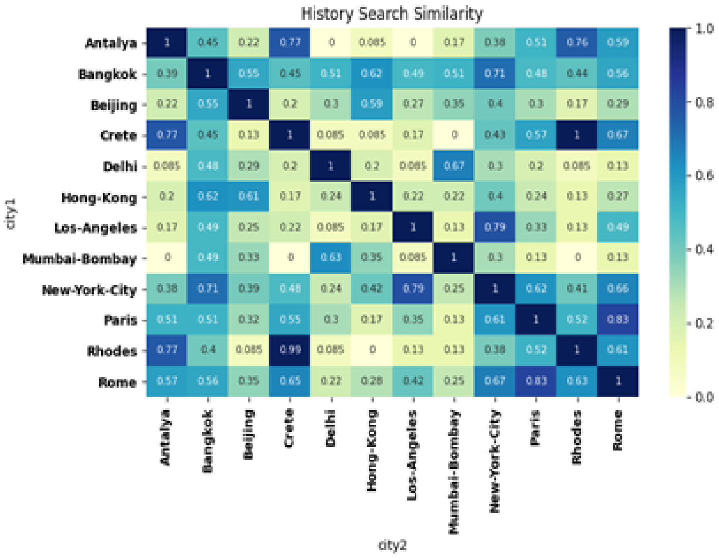
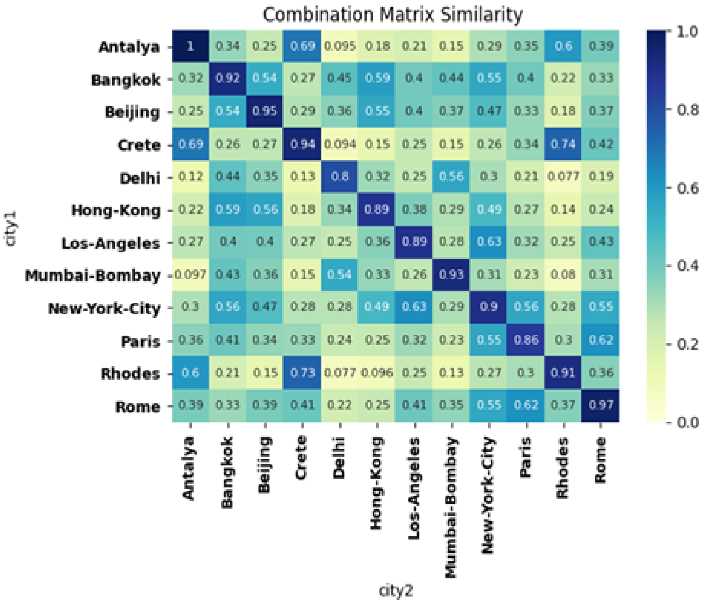


Figure 4.7: History Search

In the history search heatmap (Figure 4.7) the values were entered in log(x+1), where x is the number of searches one after the other.

§4.4. Combination Of Methods Results



	Antalya	Bangkok	Beijing	Crete	Delhi	Hong-Kong	Los-Angeles	Mumbai-Bombay	New-York-City	Paris	Rhodes	Rome
Antalya	1	0.34	0.25	0.69	0.095	0.18	0.21	0.15	0.29	0.35	0.6	0.39
Bangkok	0.32	0.92	0.54	0.27	0.45	0.59	0.4	0.44	0.55	0.4	0.22	0.33
Beijing	0.25	0.54	0.95	0.29	0.36	0.55	0.4	0.37	0.47	0.33	0.18	0.37
Crete	0.69	0.26	0.27	0.94	0.094	0.15	0.25	0.15	0.26	0.34	0.74	0.42
Delhi	0.12	0.44	0.35	0.13	0.8	0.32	0.25	0.56	0.3	0.21	0.077	0.19
Hong-Kong	0.22	0.59	0.56	0.18	0.34	0.89	0.38	0.29	0.49	0.27	0.14	0.24
Los-Angeles	0.27	0.4	0.4	0.27	0.25	0.36	0.89	0.28	0.63	0.32	0.25	0.43
Mumbai-Bombay	0.097	0.43	0.36	0.15	0.54	0.33	0.26	0.93	0.31	0.23	0.08	0.31
New-York-City	0.3	0.56	0.47	0.28	0.28	0.49	0.63	0.29	0.9	0.56	0.28	0.55
Paris	0.36	0.41	0.34	0.33	0.24	0.25	0.32	0.23	0.55	0.86	0.3	0.62
Rhodes	0.6	0.21	0.15	0.73	0.077	0.096	0.25	0.13	0.27	0.3	0.91	0.36
Rome	0.39	0.33	0.39	0.41	0.22	0.25	0.41	0.35	0.55	0.62	0.37	0.97

Figure 4.8: Combination Matrix

In the combination matrix we calculate the mean of 3 different methods:

1. text results based on 'google-encoder' results
2. images results based on the first method results
3. history search based on product-network results

§4.5. Comparing against the truth using AMT Results

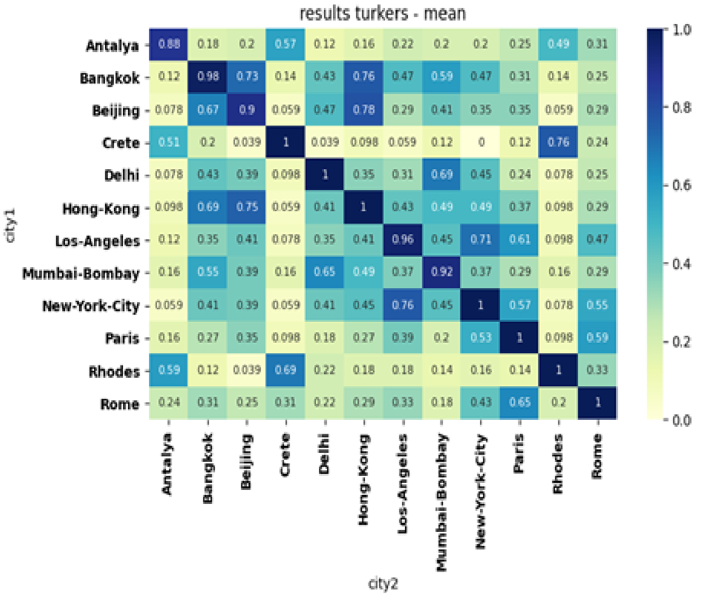


Figure 4.9: AMT results

As we explained earlier, we compared the different similarity methods against the truth using AMT results in 4 different methods:

1. In the first method, the distance between matrices are:

Results Of Method 1	
Similarity method	Score
Text	18.17602
Image Option 1	18.4473
Image Option 2	26.0520
Image Option 3	30.7578
History Search	21.5251
Combination	13.9799

In this method the most representative method is the one who gets the lowest score.As we can see that the most representative method of similarity is the measurement of similarity by combining images, text, and historical searches.

2. In the second method , bigger than threshold, the thresholds are the median of each matrix. The results are:

Results Of Method 2	
Similarity method	Score
Text	108
Image Option 1	114
Image Option 2	116
Image Option 3	107
History Search	97
Combination	116

   In this method we obtained an equation between 2 different methods of similarity, the first method is the second method of similarity between images, a method based on the mean and the standard deviation of similarity across all objectives and all images. The second method is to combine the images, text, and search history.

3. In the third method, bigger than threshold, the threshold is uniform- the median of AMT results matrix. The results are:

Results Of Method 3	
Similarity method	Score
Text	108
Image Option 1	114
Image Option 2	98
Image Option 3	101
History Search	98
Combination	114

   In this method we got an equation between 2 different methods of similarity, the first method is the first method of similarity between images, it is a method based on the amount of similarity by cos similarity between two destinations. The second method is the combination of images, text, and search history.

4. In the fourth method, bigger than threshold, the threshold is uniform- the mean of AMT results matrix. The results are:

Results Of Method 4	
Similarity method	Score
Text	113
Image Option 1	115
Image Option 2	105
Image Option 3	106
History Search	97
Combination	119

In this method we obtained that the most representative method of similarity is the measurement of similarity by combining images, text and historical searches.
As we can see from the results, the combination of the text, search history and the images(Figure 4.8) is the method that represents in the best way the people's opinions.

Chapter 5

Conclusion

Ị n this project we develop and research 4 different models for recommender flight destination in order to examine which model more represents the opinions of the people.
The models were : similarity by text , photos, product-network and combination of the models.
During the research we found that the result that represent the people's opinion is the combination of all methods together, the model based on photos gave us the best results from the individual models, after this the results based on text and the least represent model was the product-network model.
From here we can learn that product network by himself isn't sufficient enough to be a model but using him in combination with other models improve the results and gave us better(more represented) results.
From this research we can learn and see the power of the recommendation systems and the importance of the model the recommender use. From the research we can see that a combination of models together give more accurate recommendations. We can conclude from the study that that a combination of models together give more accurate recommendations , and that requires future research into different types of products in order to see if the conclusion indeed correct.

Chapter 6

References

[1] T. Deselaers and V. Ferrari. Visual and semantic similarity in ImageNet. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 1777–1784. IEEE, 2011.

[2] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In Advances in Neural Information Processing Systems, pages 658–666, 2016.

[3] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th International Conference on World Wide Web, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[5] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 43–52. ACM, 2015.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[7] M. Tan, S.-P. Yuan, and Y.-X. Su. A learning-based approach to text image retrieval: using cnn features and improved similarity metrics. arXiv preprint arXiv:1703.08013, 2017.

[8] G. W. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 2729–2736. IEEE, 2011.

[9] G. Wang, D. Hoiem, and D. Forsyth. Learning image similarity from flickr groups using stochastic intersection kernel machines. In Computer Vision, 2009 IEEE 12th International Conference on, pages 428–435. IEEE, 2009.

[10] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In Visual Analytics Science and Technology, 2006 IEEE Symposium On, pages 191–198. IEEE, 2006

[11] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics, 2:67– 78, 2014.

[12] Image Based Fashion Product Recommendation with Deep Learning Hessel Tuinhof1, Clemens Pirker2 , and Markus Haltmeier3

[13]- Hoo-Chang S, Roth HR, Gao M, Lu L, Xu Z, Nogues I et al (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 35(5):1285
[14]- L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han and T. S. Huang, "Aworldwide tourism recommendation system based on geotaggedweb photos," 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 2274-2277, doi: 10.1109/ICASSP.2010.5495905.
[15]- M. G. Ozsoy, "From word embeddings to item recommendation", CoRR, vol. abs/1601.01356, 2016, [online] Available: http://arxiv.org/abs/1601.01356.
[16]- H. Gao, J. Tang, and H. Liu, “Exploring social-historical ties on location-based social networks,” in Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012, 2012
[17]- A Comparison of Product Network and Social Network Based Recommendation Engines for Twitter Users Shawndra Hill Adrian Christophe Van den Bulte.
[18]- M. Shang, Y. Fu and D. Chen, "Personal Recommendation using Weighted Bipartite Graph Projection," 2008 International Conference on Apperceiving Computing and Intelligence Analysis, 2008, pp. 198-202, doi: 10.1109/ICACIA.2008.4770004.