DATA DRIVEN EXPLORATION OF DOCUMENT COLLECTION TO UNDERSTAND UNDERLYING SOCIAL FABRIC USING GRAPH REPRESENTATION LEARNING

by
ANIL OZDEMIR

Submitted to the Graduate School of Engineering and Natural Sciences in partial fulfillment of the requirements for the degree of Master of Science

Sabancı University
July 2021

DATA DRIVEN EXPLORATION OF DOCUMENT COLLECTION TO UNDERSTAND UNDERLYING SOCIAL FABRIC USING GRAPH REPRESENTATION LEARNING

Approved by:

[redacted]

[redacted]

[redacted]

Date of Approval: July 2, 2021

ABSTRACT

DATA DRIVEN EXPLORATION OF DOCUMENT COLLECTION TO UNDERSTAND UNDERLYING SOCIAL FABRIC USING GRAPH REPRESENTATION LEARNING

ANIL OZDEMIR

Computer Science and Engineering, Master's Thesis, July 2021

Thesis Supervisor: Prof. Selim Balcısoy

Keywords: Graph Theory, Natural Language Processing, Machine Learning, Graph Representation Learning , Recommendation Systems

An enormous collection of documents is digitally available in text, images, and other representations for cultural heritage (CH). The availability of such extensive data creates a need for various approaches that allow users and archivists to understand latent relationships in collections. However, one of the biggest challenges of documents in cultural heritage is that it takes a long time and is difficult for archivists to analyze and process documents. Due to this manual process, there may be situations where the person, place, and events mentioned in these documents are not expressed in the same linguistic terms and words, or they contain ambiguous concepts that make it difficult to understand; as a result, it is challenging to uncover these relationships without careful examination by a professional. Therefore, there is a need for an archivist who will re-analyze these terms to capture similar events, persons, and places between the documents and thus reveal the latent relationship. To fill this gap, we proposed a system that combines various NLP algorithms and graph representation learning methods using only the textual summary of the documents and the documents' metadata. The system automatically extracts substantial terms in the documents, then produces embedding for the documents themselves and these terms. Finally, the proposed system has been used to explore the document collection and perform document recommendations by utilizing calculated document embeddings. We evaluated and compared the performance of the proposed work with alternative methods through an experiment we conducted with archive experts.

ÖZET

DOKÜMAN KOLEKSIYONU ÜZERINDE GRAFIK TEMSIL ÖĞRENIMI KULLANARAK TEMELDEKI SOSYAL YAPIYI ANLAMAK

ANIL ÖZDEMIR

PROGRAM ADI YÜKSEK LİSANS TEZİ, MAYIS 2021

Tez Danışmanı: Prof. Dr. Selim Saffet Balcısoy

Anahtar Kelimeler: Heterojen Bilgi Ağları, Grafik Teorisi, Doğal Dil İşleme , Makine Öğrenmesi, Grafik Temsili Öğrenme, Öneri Sistemleri

Kültürel Miras (CH) alanında metin, resim ve diğer temsil türlerinde dijital olarak çok büyük bir belge koleksiyonu ortaya çıkmaya başlamıştır. Bu tür kapsamlı verilerin mevcudiyeti, araştırmacıların ve arşivcilerin koleksiyonlardaki gizli ilişkileri anlamalarına olanak tanıyan çeşitli yaklaşımlara ihtiyaç duymaktadır. Ancak kültürel mirasta belgelerin en büyük zorluklarından biri, arşivcilerin belgeleri analiz edip işlemesinin uzun zaman alması ve zor olmasıdır. Bu manuel işlem nedeniyle, bu belgelerde adı geçen kişi, yer ve olayların aynı dilsel terim ve kelimelerle ifade edilmediği veya anlaşılmasını zorlaştıran belirsiz kavramlar içerdiği durumlar olabilir; sonuç olarak, bir profesyonel tarafından dikkatli bir şekilde incelenmeden bu ilişkileri ortaya çıkarmak zordur. Bu nedenle, belgeler arasındaki benzer olayları, kişileri ve yerleri yakalayacak ve böylece gizli ilişkiyi ortaya çıkaracak bu terimleri yeniden analiz edecek bir arşivciye ihtiyaç vardır. Bu boşluğu doldurmak için, yalnızca belgelerin metinsel özetini ve belgelerin ekstra meta-verilerini kullanarak çeşitli doğal dil işleme algoritmalarını ve grafik temsili öğrenme yöntemlerini birleştiren bir sistem önerdik. Sistem otomatik olarak belgelerdeki önemli terimleri çıkarır, ardından belgelerin kendileri ve bu terimler için vektörel değerler üretir. Bu vektörel değerler kullanılarak dokümanlar arasındaki ilişkileri gözlemlemek ve ilgili dokümanları ortaya çıkarmak kolaydır. Son olarak önerilen sistem, hesaplanan vektörler kullanılarak doküman önerisi yapmak için kullanılmıştır. Önerilen çalışmanın performansını alternatif yöntemlerle arşiv uzmanlarıyla yaptığımız bir deneyle değerlendirdik ve karşılaştırdık.

ACKNOWLEDGEMENTS

It is a genuine pleasure to express my sincere gratitudes to my mentor and thesis supervisor Prof. Selim Saffet Balcısoy for his permanent support, patience and understanding throughout the past five years.

I owe a deep sense of gratitude to Dr. Onur Varol for his keen interest on me at every step of my research. His prompt inspirations, timely suggestions with kindness , enthusiasm and dynamism have enabled me to complete my thesis.

I would like to express my sincere gratitudes to Dr. Reyyan Yeniterzi for the technical background she gave me in the field of natural language processing and her presence in the thesis jury.

In addition, I would like to thank Prof. Burcin Bozkaya for his invaluable comments and feedbacks on the study and his presence in the thesis jury.

I thank Emir Alaattin Yılmaz , Hasan Alp Boz and Mert Gürkan for the great times in the lab. And my sincere thanks goes to to all my friends and participants who helped me in my experiment for research. It is my privilege to thank my brother Dr. Alp Özdemir, for his constant encouragement throught my entire life and research period.

And finally, I would like to thank my family for their support throughout my entire life.

To all my dear friends and family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1. Introduction

Today, almost all cultural heritage (CH) institutions are starting to digitize parts of their collections and archives to improve accessibility, preservation of originals, publicity, and visibility of the institution on the Internet. With this recent development, digital document collections have been multiplying. These collections are spread over more than one area of life in a vast domain, including art, history, mathematics, physics, etc. Such a situation creates a substantial volume of documents digitally available. The increasing demanding for digitalization in the cultural heritage domain also derives several challenges for the analysis of documents in this field. One of the first challenges is reading, analyzing, and processing the documents takes lots of time and requires personnel and funding. During such processing, there may be situations where the important entities in these documents are not expressed in the same linguistic terms and words, or they may contain ambiguous concepts that make it difficult to understand; as a result, it is challenging to uncover these relationships without careful examination by experts and the amount of information that is overlooked in the manual analysis is too much. These challenges create the need for various approaches that allow users to understand latent meanings in collections, discover and investigate relationships, and extract the necessary information from collections in CH.

In the literature, studies have been carried out to solve the challenges mentioned in the field of CH and to assist archivists, and these studies have benefited from methods from many different fields. The majority of these methods use natural language processing (NLP) techniques, including Named Entity recognition (Nadeau & Sekine, 2007), to expose mementos that specify who has mentioned the notions, events, and concepts in the collection take place. Another example is topic modeling techniques like Latent Dirichlet Allocation (Blei, Ng & Jordan, 2003), and Latent Semantic Analysis (Dumais, 2004) to understand the underlying theme, exhibit periods, dates, and expose such temporal expressions from textual collections. Simultaneously, there are many studies using visualization tools and interfaces to deal with the same problem. Examples of these include MindMap (Spangler, Kreulen &

Lessler, 2002) that using visualization and utilizing taxonomies to understand textual collections, InfoTouch (Kristensson, Arnell, Björk, Dahlbäck, Pennerup, Prytz, Wikman & Åström, 2008) that explorative visualization interface for photo collections. Along with the improvements mentioned in collection and document research, we turn our attention to improving corresponding techniques in the context of CH domains.

In this thesis, two different systems were introduced using a collection of documents in the field of cultural heritage. To access such a collection, we have collaborated with archive professionals from a cultural institution, SALT (`saltonline.org`) which focused on public service producing research-based exhibitions, publications, and digitization projects. The first proposed study is a visual exploration tool that uses NLP methods to make it easier to explore documents in the CH area which is described in Chapter 3. In the proposed visual tool our contribution is design and development of a visual exploratory tool that facilitates the uncovering of hidden information and stories underlying documents, extracting the key individuals, temporal expressions, locations, entities, and keywords within the documents, and establishing a network between documents using both NLP and visualization methods from archive metadata. Such visualization design will allow researchers and archivists to form and test hypotheses and observe individual relationships, networks, semantic and syntactic proximity, and stories present in the archives' metadata collections.

In the second part which is described in Chapter 4 and 5, we introduced a system that combines various natural language processing algorithms and heterogeneous network embedding methods using only the textual summary of the documents and the document's metadata. The system extracts n-grams that describe linguistically and combines them with temporal information and extra categorical information that exists in data to represent them on a heterogeneous graph network. Then, the system produces vector representations for the documents in the collection by utilizing graph representation learning algorithms, and we used these representations both in visualization and in design a document recommendation system. It is easy to observe the relationships between the documents and reveal related documents by using these vector embeddings. Finally, we experimented with archival professionals to test the performance of the recommendation system we produced and showed that our system outperformed the other models we compared.

The contributions of the thesis we have presented are as follows.

- A visual tool has been developed that utilizes current natural language processing methods to explore documents in the cultural heritage area. This tool helps archivists to quickly observe documents and see important people, terms, and locations contained in documents.

- There is a lack of a document recommendation system in a collection of documents in the field of cultural heritage in the literature, and we have developed a recommendation system to fill this gap. Since the documents in this field are difficult for a normal person to read and understand in languages, state-of-the-art graph representation and natural language processing methods have been used to make it easier to analyze the language of these documents.

- The recommendation system utilizes underlying n-grams that describe linguistically and combines them with temporal information and extra categorical information that exists in data to represent them on a heterogeneous graph network. Then, use the network structure to obtain meaningful embeddings of documents and use embeddings to recommend documents to a user. Since the user's previous preferences are not used in this recommendation system, the introduced system is also effective in solving the cold-start problem, which is described in Chapter 2.4.1.

- Manual annotation experiment with cultural heritage experts demonstrates the success of our system against alternative methods.

- An example of data-driven analysis of cultural heritage collection is presented, helping to understand the social fabric of a Crete in the 19th Century.

The remaining of the thesis is structured as follows: in Chapter 2, a broad literature review is presented based on the underlying methods and algorithms. The designed visual exploratory analysis tool is explained in Chapter 3. The underlying methods we benefited from and the implementation design for the document recommendation system we have introduced explained and described in 4. Chapter 5 describes the data we use when developing and testing the recommendation system and shows the experimental design and results of the experiment we conducted to measure the performance of the system we presented. Finally, Chapter 6 consists of the conclusion and future work.

Chapter 2. Related Work

2.1 Exploratory Visual Interfaces in Cultural Heritage

There has been a significant effort in digitization and CH preservation, visualization, and interaction. Recently, most of this work has been focused on creating digital representations of cultural artifacts and the creation of metadata and documentation associated with this. This effort's significant consequence is that there is an enormous collection of documents digitally available in text, images, and other representations. An example of such multidisciplinary collections includes the Biodiversity Heritage library (Gwinn & Rinaldo, 2009) which is an open-access digital library of 12 natural history collections, a digital archive of the Ottoman/Turkish serial novel, which is one of a broader digital archive that preserves, classify and analyze the cultural heritage of Ottoman/Turkish society (Serdar & Tutumlu, 2018) and an Arts and Humanities Data Service (AHDS) (Burnard & Short, 1994) which is a UK national service aiding the discovery, creation, and preservation of digital resources that cover the areas, such as archaeology, history, literature, and linguistics.

The increase in the number of these archives and collections, visualization, exploration, and relationship extraction projects using such digital collections has increased considerably in the last 20 years. Examples of these include EPOCH (Petridis, Pletinckx, Mania & White, 2006) that is a multimodal interface for interacting with digital heritage artifacts by using virtual reality, CULTURA (Hampson, Agosti, Orio, Bailey, Lawless, Conlan & Wade, 2012) which is a project that covers topics, such as finding entity, event, and relationship extraction within unstructured text obtained from digital CH collections using various NLP techniques. Dou, Qin, Jin & Li (2018) proposes a knowledge graph to finding ontology and relationship for intangible cultural heritage (ICH). Moreover, Aviles Collao, Diaz-Kommonen, Kaipainen & Pietarila (2003) describes tools that produce similarity cluster representations in CH content using spatial vector-based computation technique, and Salisu, Mayr, Filipov, Leite, Miksch & Windhager (2019) introduce visual techniques to depicting development and changes over time in CH collections.

In other domains beyond CH, there are enormous visualization and NLP methods to represents and discover collections. For example,Cheng, Wang, Huang, Chundi & Song (2020) enhances the collection understanding by presenting a topic-modeling-based text visualization approach that extracts and visualizes topic models regarding biomedical document collections. In addition, Chen, Zou & Scott (2016) developed a topic-based visualization model to present and analyze the Illinois Digital Archives, in which a document can now be represented as a mixture of semantic topics and – cite2vec – is a new visualization scheme that allows the user to dynamically explore and browse documents by embedding documents into a high-dimensional space by using word2vec (Mikolov, Sutskever, Chen, Corrado & Dean, 2013). The relationship between documents is based on both distances between document vectors and shared attributes between documents in the proposed tool. On the other hand, recent approaches usually model the documents through its underlying theme by examining the set of words comprising the document using Topic models (Cheng et al., 2020),(Chen et al., 2016). These kinds of models usually discover the abstract “topics” that occur in collecting documents by using statistical properties of word counts. However, for users who need to explore and discover documents as part of the research process, such a model may fail to find a relationship between documents.

2.2 Representation Learning for Graph Networks

Representation learning depicted in Figure 2.1 and stands for learning the underlying representations of the data that faciliate extracting beneficial information when constructing machine learning models including classifiers or other predictors. Bengio et al. (2013) describes good representation as the one that captures the posterior distribution of the the underlying explanatory factors for the observed input. In recent years, representational learning studies have grown rapidly across different fields in the literature and industry. Examples of these including speech recognition and signal processing (Dahl, Ranzato, Mohamed & Hinton, 2010),(Deng, Seltzer, Yu, Acero, Mohamed & Hinton, 2010) ,(Chorowski, Weiss, Bengio & van den Oord,

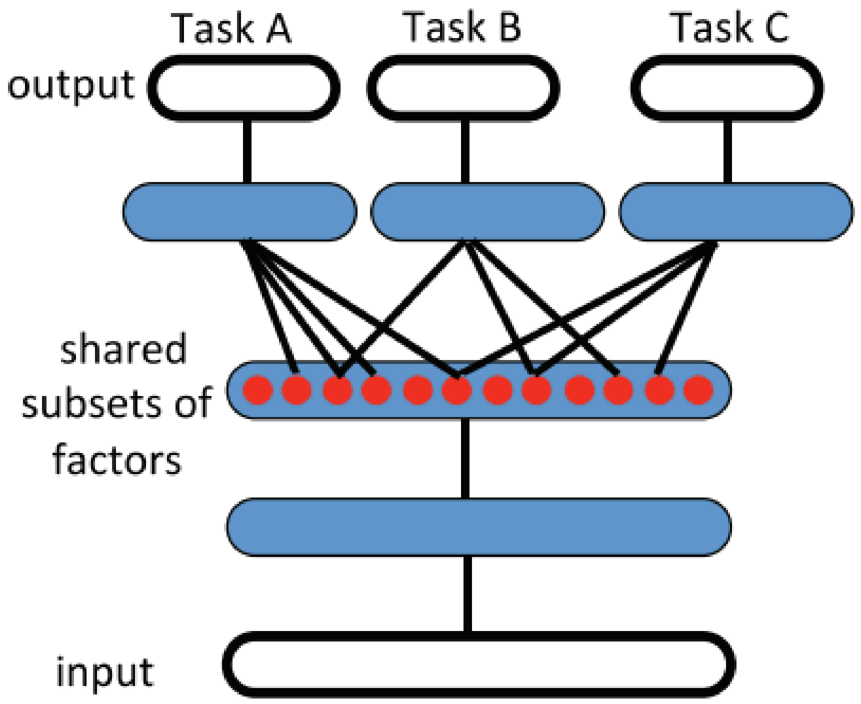


Figure 2.1 Illustration of representation-learning discovering explanatory factors (middle hidden layer, in red), some explaining the input (semi-supervised setting), and some explaining target for each task. Because these subsets overlap, sharing of statistical strength helps generalization as Bengio et al. (2013) argues

2019)), object recognition (Wang, Sun, Cheng, Jiang, Deng, Zhao, Liu, Mu, Tan, Wang & others, 2020),(Gidaris, Singh & Komodakis, 2018) , natural language processing (Collobert, Weston, Bottou, Karlen, Kavukcuoglu & Kuksa, 2011),(Liu, Lin & Sun, 2020),(Mikolov, Grave, Bojanowski, Puhrsch & Joulin, 2017),(Tschannen, Bachem & Lucic, 2018).

The major challenge in the machine or deep learning on graphs is finding a way to combining information about graph structures into a machine learning model and therefore the same challenge applies to representational learning. Such challenge is explained by Hamilton et al. (2017) as one could encode the pairwise properties between nodes, such as a number of shared friends or strength of the relationship in the case of link prediction in social network-based graphs. Based on this situation, there is no easy way to encode this multi-dimensional, non-Euclidean information about graph structure into a feature vector. Studies done in the literature to extract structural information from graphs are generally based on summary graph statistics (Bhagat, Cormode & Muthukrishnan, 2011), kernel functions (Vishwanathan, Schraudolph, Kondor & Borgwardt, 2010) or carefully hand-engineered features (Liben-Nowell & Kleinberg, 2007). However, such approaches are limited and not efficient in time but also expensive, and this situation has made it inevitable to use representation learning on graphs.

Recently, several approaches have been proposed to learn the representation that encodes the structural information depicting graphs. The principal idea shared by these different approaches is to learn mapping function f that embeds nodes, links, or entire (sub)graphs, as points in a low-dimensional vector space R^d. The objective of the mapping function is to optimize this mapping so that geometric relationships in the vector space reflect the topological structure of the original graph and illustrated in Figure 2.2. The embeddings that are optimized as a result of such representation learning can be used later in many downstream machine learning tasks including classification, clustering, similarity search Hamilton et al. (2017). Representation learning with graph-structured information based works includes combination of latent and observable relational learning models on graphs in (Nickel, Murphy, Tresp & Gabrilovich, 2015) , Latent space approaches to social network analysis (Hoff, Raftery & Handcock, 2002) and geometric deep learning on graphs (Monti, Boscaini, Masci, Rodola, Svoboda & Bronstein, 2017) , (Bronstein, Bruna, LeCun, Szlam & Vandergheynst, 2017).

Graph representation learning also under the topic of graph-based subspace learning. Previous studies include Locally Linear Embedding (Li, Lin, Yan & Xu, 2008), Laplacian Eigenmap (Belkin & Niyogi, 2001) . The underlying idea of these methods to construct a proximity graph as an approximation to the underlying data and then learn a low dimensional vector representation for the data by preserving proximity graph-based structure. Thus, one could calculate the similarity between two arbitrary objects with the inference from learned space. Traditional graph-based subspace learning algorithms only take into account one type of object, while Metapath2vec algorithm that used in this study deals with multi-type interrelated data objects.

2.3 Random Walk Based Graph Network Embedding Methods

Traditionally, most of the methods for extracting latent space for graph-structured information based on solving eigen decomposition, and the complexity starts to increases quadratically as the number of vertices increases. Such approaches are inefficient in solving large-scale graph problems. Such shortcoming is the reason behind the random walk strategy-based representation learning algorithms on graphs gained momentum and became the basis of the metapath2vec (Dong, Chawla & Swami, 2017) method used during this study.

In the literature, the first method among the methods that combine the random walk strategy and skip-gram (Mikolov, Chen, Corrado & Dean, 2013) to learn latent representations is Deepwalk (Perozzi, Al-Rfou & Skiena, 2014), which generates uniform random walks from graphs to learn representations and treating walks as the equivalent of sentences in Word2vec. In other words, DeepWalk adopted the idea underlying the Wordvec (Mikolov et al., 2013) , which was a breakthrough in natural language processing. In word2vec, input words that have similar context have expected to have proximate positions in vector space and this corresponds to nodes that have similar neighborhoods that will acquire similar vector representations in Deepwalk. Such a neighborhood is described as a sliding window across the generated random walk and defining all the nodes within a particular window as context for the center node of the window. Thus, DeepWalk approximates the conditional probability of alternative vertice in the network being close by optimizing for high probabilities of vertices in the neighborhood.

LINE (Tang, Qu, Wang, Zhang, Yan & Mei, 2015) is one of the first studies in the random walk approach for learning representations of vertices in a network and argues that DeepWalk does not provide a clear objective that expressing what type of network properties are preserved. Also, LINE states that DeepWalk expects nodes with higher second-order proximity produces similar representations, while the LINE preserves both first-order and second-order proximities by optimizes the representations to capturing both first and second-order proximity by training different embedding vectors for them and concatenate later.

Later, Node2vec is proposed by Grover & Leskovec (2016) and generalizes the DeepWalk and LINE by introducing parameters to controlling random walk behavior and argues that added flexibility in exploring neighborhood is helping learn richer vector representations using Breadth-First and Depth-First sampling. Node2vec performs better results over state-of-the-art techniques on that time on multi-label classification and link prediction in several real-world networks from different domains. HARP (Chen, Perozzi, Hu & Skiena, 2018), and Wallets (Perozzi, Kulkarni, Chen & Skiena, 2017) can be given as an example of extra studies that present graph embedding algorithms using the random walk strategy.

The methods are given earlier focused on homogeneous networks in general, and Dong et al. (2017) highlighted this shortcoming and introduced the Metapath2vec which studies representation learning in heterogeneous networks. Dong et al. (2017)

stated that conventional graph embedding techniques are limited when multiple types of nodes and links exist. The proposed Metapath2vec model modifies the random-walk strategy by combining walks with the meta-path-based concept to construct a heterogeneous neighborhood of vertices and then utilize a heterogeneous skip-gram model to extract embeddings. In this way , Metapath2vec helps to capture both semantic and structural relations in heterogeneous networks.

The reason behind using metapath2vec in this study is that the graph structure we construct with the documents and the features we extracted from the documents is a heterogeneous graph, and we determined that metapath2vec is the most accurate option to capture the relations between different node types both structurally and semantically.

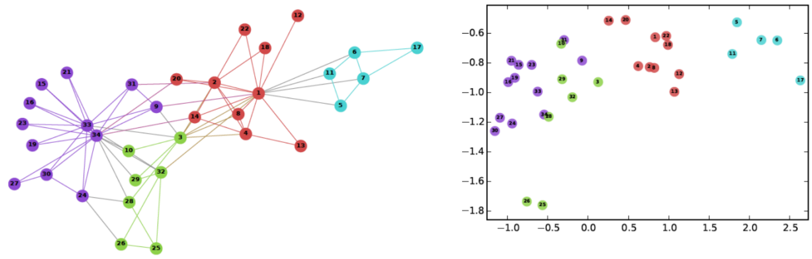


Figure 2.2 Graph Representation Learning Illustration. Network space is depicted on the left and vector space is depicted on the right, retrieved from Hamilton et al.

2.4 Representation Learning for Recommendation Systems

In this section, we will focus specifically on the development of document recommendation systems, recommendation systems designed for documents in the field of cultural heritage, and the use of representation learning techniques in recommendation systems. A Recommendation system defined a subclass of information filtering design that tries to predict the "rating" or "preference" a user would give to a specific item (Ricci, Rokach & Shapira, 2011). As discussed in Section 2.2, the rapid increase in the number of studies in the field of representation learning has also led to the use of these algorithms in recommendation systems. Recently, recommendation systems have been designed for many different industries and domains including video and music services (Van Den Oord, Dieleman & Schrauwen, 2013) , product recommendation for online shopping sites (Lee & Kwon, 2008) , social content recommendation for social media platforms (Wang, Zhu, Cui, Sun & Yang, 2013) , hotel or tour recommendation for tourism industry (Yu & Chang, 2009) , news recommendation (Zheng, Zhang, Zheng, Xiang, Yuan, Xie & Li, 2018) and document recommendation (Weng & Chang, 2008).In the past, common approaches for recommendation systems are collaborative filtering approach (CF) which uses shared characteristics of users to use in the recommendation, and (Sarwar, Karypis, Konstan & Riedl, 2001) the content-based approach (CB) (Van Meteren & Van Someren, 2000) which based on content and relevant profiles to make recommendations to users. CB also benefits from past interactions and behaviors to make successful recommendations.

2.4.1 Cold-start problem in Recommendation Systems

The cold-start problem is defined by Maltz & Ehrlich (1995) and states the important issue at the initial learning stages of the recommendation systems. The issue is that in the initial stage, there is a lack of information about the user, or it is difficult to gather records about users, but still a recommendation service must be provided to the users. Since the mentioned information is critical to the operation of the recommendation systems, if the relevant information is not available, it produces a cold-start problem and reduces the effectiveness of the recommendation systems. As a result, users' desire to use these systems also decreases. As Weng & Chang (2008) argues, there are two cold-start problems as follows:

- *New-system cold-start:* In its initialization stage, the new system still has no information for evaluation and recommendation from users of the items. Thus, the system lacks related information and characteristics about users.
- *New-user cold-start:* Recommendation system works for a couple of sessions and has also aggregates and stores a fragment of user profiles as well as evaluation archives. However, for the newcomer user, the system is still unable to seek useful records for the recommendation, thus, it might not satisfy the needs of newcomer users.

For the recommendation mechanism, whether it is a CF-based or CB-based recommendation, there have still been difficulties in handling the cold-start problems. Although, no certain best solution for improvement, many studies have argued that the hybrid-based approaches may constitute a better recommendation result.

2.4.2 Document Recommendations

One of the first proposed works for document recommendation was a Fixit system (Hart & Graham, 1997) , which is defined as query-free search and, supply the benefits of a combined expert and text database system without requiring query-procedure knowledge to the user. Another early-stage document recommendation work proposed by Budzik & Hammond (2000) and developed a system that helps users with recommending relevant documents while browsing the Web. In the proposed work, queries and documents are represented as vectors in a high dimensional space, where each dimension represents a word. Among more recent examples, Weng & Chang (2008) use ontology and the spreading activation model for research paper recommendation. Authors utilize ontology to construct a profile for users and benefit from user profile ontology as the basis to reason about the needs of users. Authors expressed the spreading activation model in a format of network data structure where nodes represent the characteristics of objects and edges express the relationships between objects and argues from conventional recommendation systems suffer from the cold-start problem (see Section 2.4.1). Nagori & Aghila (2011) proposed hybrid model that uses Latent Dirichlet Allocation (Blei et al., 2003) and CF-based techniques to perform document recommendation that chooses top K document for users. Shaparenko & Joachims (2009) proposed a method that utilizes language modeling and convex optimization to recommend documents. Authors unsupervised generative model for large text corpora that provides a formal structure for the process and recommend the top-N most similar documents according to the cosine similarity.

2.4.3 Representation Learning Applications in Document Recommendation Systems

The effort of representation learning in document recommendation systems has become very common especially in the last 5 years, but one of the first proposed studies about representation learning applications in this domain is a novel graph-based representation learning algorithm designed for document recommendation in social tagging services (Guan, Wang, Bu, Chen, Yang, Cai & He, 2010). In the work, the authors reformulate the document recommendation problem into a graph-based representation learning framework and present a novel algorithm called *Multi-type Interrelated Objects Embedding* and try to represent users and documents in the same vector space in which two related objects are close to each other. In the experiments, the proposed method outperforms traditional recommendation algorithms.

Afterward, Yang, Liu, Zhao, Sun & Chang (2015) proposed a text-associated network representation method that incorporates text features of vertices into graph representation learning using matrix factorization. De Boom, Van Canneyt, Demeester & Dhoedt (2016) presented representation learning framework for short texts using a weighted word embedding aggregation and argue that method is capable of holding the semantic information in the texts, and is applicable out-of-the-box. Gupta & Varma (2017) addressed the problem of scientific paper recommendation by proposing a novel method using distributed representations of texts and graphs. Authors combine network embeddings with the semantic embeddings of the texts. Another study conducted by Zhang, Ai, Chen & Croft (2017) and proposed a joint representation learning framework for top-N recommendation with heterogeneous information sources including texts. The authors aim to capture the word sequential information and their local semantics to better modeling of the textual reviews for a recommendation. VOPRec (Kong, Mao, Wang, Liu & Xu, 2018) is a scientific paper recommendation system that uses text information and structural identity. Text information is represented with word embedding to find top-N similar scientific papers. Authors combine the Doc2vec embeddings with Struc2vec embeddings to reconstruct the paper network and apply matrix factorization to learn graph structures and construct paper vectors. Finally, proposed system recommends top-n paper according to the cosine similarity. Brochier, Guille & Velcin (2019) presented a GVNR, a matrix factorization-based method for network embedding. The authors use textual content associated with the nodes to learn representations of words and documents and stated that the proposed system demonstrates state-of-the-art results on a wide range of networks and able to produce recommendations based on the distance between words, documents, and embeddings.

Chapter 3. A NLP Enhanced Visual Analytics Tool for Archives Metadata

3.1 Design Rationale

Our developed exploratory tool aims to help archivists form and test hypotheses and observe certain relationships, networks, semantic and syntactic proximity, and stories present in the metadata archives. To achieve this goal, the extracted features from metadata must be displayed in various aspects comprehensively. These features include features in different domains that can best describe archives. We have divided this attribute pool into two, the ones we have obtained by ourselves using NLP methods, and those already present in the metadata. Previously available features include the year the document was published, the document's location, and the topic of the document. On the other hand, the attributes we extract with NLP and morphological methods are local locations, persons, dates, similarity, entities, keywords. We have integrated all of these attributes into the visual tool we have prepared to help an archivist analyze a document in the best possible way.

3.2 System Description

3.2.1 Feature Engineering and Transformation

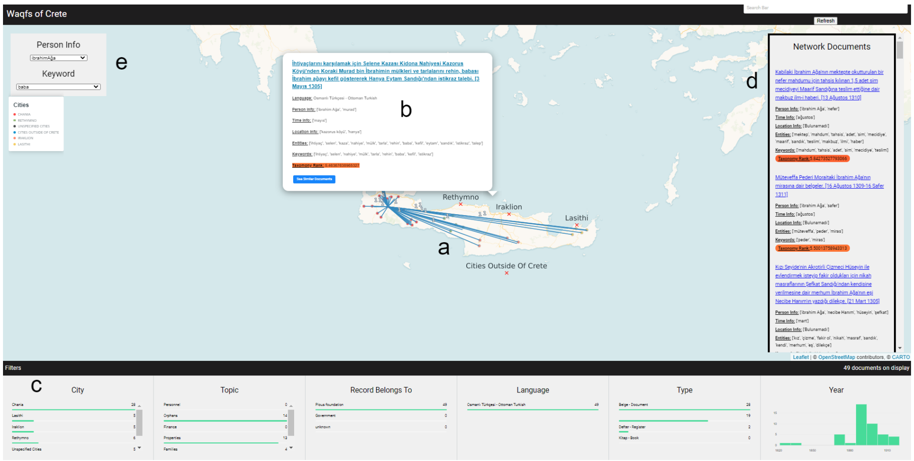


Figure 3.1 : The designed user interface. a) Interactive map. b) Information box, document-specific attributes were placed on this box. c) The filtering panel. d) The document browser that integrated enables users to browse documents easily. e) Individual and keyword search menu.

3.2.1.1 Information Extraction

From the raw metadata, we extracted the following features:

Location, Time: In this paper, we refer to villages and settlements as locations. To obtain them, we extracted location-expressing words morphologically using regex (Aho & Corasick, 1975). Although we have the year each document was published, the concept of time in the document's depiction describes a time progression, such as morning, evening, yesterday, the next day.

Entities, keywords, and key individuals: For extracting entities, we first located nouns in the text and then determine what type of entity they refer to – such as a person, location, and keywords. To perform accurate morphological analysis on textual data we used morphological parser and disambiguation parser which was proposed by Sak, Güngör & Saraçlar (2008). Proposed Parser receives complete text as an input and adds the meaning of the words into the equation while analyzes and disambiguates the words. Since a word in the Turkish language has more than one morphological analysis output, the meanings of the words in the sentence directly affect these results. In this context, disambiguation is a very crucial step for morphology and the success of the disambiguator proposed by Sak et al. (2008) has been reported as 96.45% on a disambiguated Turkish corpus. To highlight and find key individuals, since the period used in the data belongs to the period before the surname law was issued, we first captured the human names with the help of regex and morphological analyzer and then combined them with the titular adjectives. For keywords, they selected among the entities used in specific proportions.

Similarity Between Documents: In order to capture similarity between documents ,two state-of-the-art word embedding models which are Word2vec (Mikolov et al., 2013) , FastText (Bojanowski, Grave, Joulin & Mikolov, 2016) trained over our corpus. Also, we used sentence transformer (Reimers & Gurevych, 2020) which provides a method to compute dense vector representations for sentences and paragraphs using recently released pre-trained Turkish Bert model(Schweter, 2020). Consequently, each document was represented as fixed-sized mathematical vectors, and the relationship between documents was calculated by taking the cosine similarity which corresponds to these vectors' inner products.

3.2.1.2 Network Extraction

Network: We have built a homogeneous network between documents by using individuals and entities we extracted. In the network, nodes correspond to the documents themselves. To assign an edge between two documents in the network, the number of shared individuals and keywords between documents is computed. The majority of the documents do not have frequent individuals and keywords. To prevent cluttered edges and misleading results, edges are set based on a threshold value. In other words, if the number of shared individuals or keywords between two documents is higher than the predetermined threshold value, an edge is drawn between them.

3.2.2 Visual Interface and Components

In Figure 3.1, the designed interface, consisting of six components, can be observed. The main component of the interface is the interactive map Figure 3.1-a in which documents on the data, location information are integrated to describe the events at different points on the island of Crete. Here, the user can also view documents outside the island of Crete using the embedded zoom functionality. The user can navigate the documents placed according to the latitude and longitude information on an interactive map. Whenever the user needs to see extra information on the document, they can click on it and access the information box depicted in

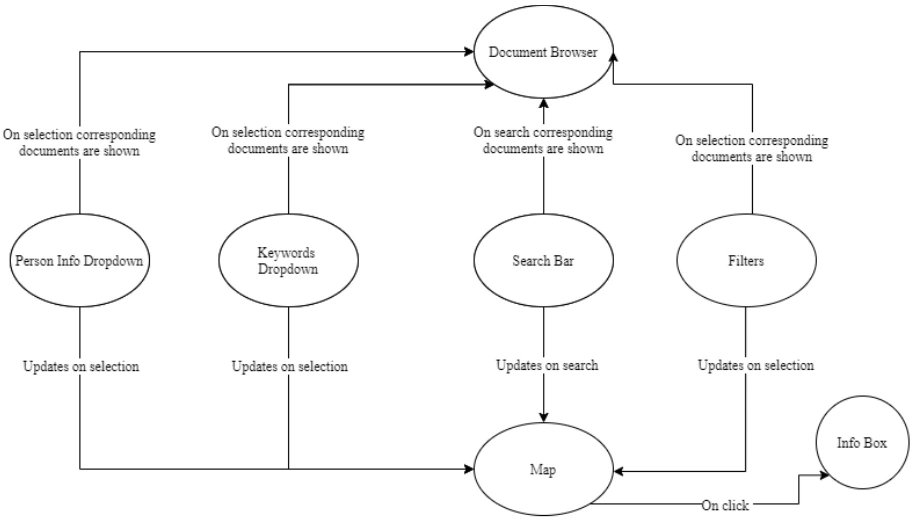

Document Browser
On selection corresponding documents are shown
On selection corresponding documents are shown
On search corresponding documents are shown
On selection corresponding documents are shown
Person Info Dropdown
Keywords Dropdown
Search Bar
Filters
Updates on selection
Updates on selection
Updates on search
Updates on selection
Map
Info Box
On click

Figure 3.2 Interaction schema among the components

Figure 3.1-b. Document-specific attributes such as location, time, person, entities, and keyword extracted using various NLP techniques and models were placed on this information box, as stated in 3.2.1.2. Another significant feature of this information box is to provide the user with documents similar to the document being observed with a button placed under the bar. After clicking the mentioned button, similar documents were computed by measuring the distance between the document-specific vectors extracted from word embedding models. Here, the documents most similar to each document were shown to the user via the document browser in Figure 3.1-d.

The panel is shown in Figure 3.1-d is the document browser that is integrated to enable users and researchers to browse documents easily. This browser has been designed to display two different system outputs on it. The first is to show the user the most similar documents to the selected document, sorted by similarity score, while another objective is to show other documents in the selected document's network diagram created by using shared keywords and key individuals. Thus, the browser's results at that moment, depending on the user's current operation. Another core component of the application is the person and keyword search menu shown in Figure 3.1-e. Through this menu, the user is given the option to select the keyword and person, which is the NLP models' output, and the documents containing these keywords or individuals are visualized on the map. The output of this action reflects the documents filtered according to the selected person and keywords on the map, and the user can click on the documents and access their information boxes. The part that differs from the previous information box is that the user can view the network created for that document on the map and the document browser with the "network" button on this information box. The network formed here includes documents that contain the same keyword or person as the person and keywords contained in the selected document, as mentioned in the 3.2.1.2 section. In the last component of the interface, the filtering panel is shown in Figure 3.1-c and located at the bottom of the tool; the user has been allowed to filter the documents currently displayed on the map with some extra attributes in the document's metadata. These attributes are the city where the document is located, the topical of the document, the document's language, the type of the document (image, document, etc.), and the year the document was published.

Chapter 4. Recommendation System Using Metapath2vec

4.1 Architecture

In this chapter, we will present a study where we create a heterogeneous network structure using raw texts and the features we extract using the metadata of these texts, obtain node embeddings by leveraging state-of-the-art graph representation methods, and use these embeddings for both recommendation and exploration. The theory behind the system presented in section 4.2 of this chapter is explained, while in section 4.3 our implementation details and design are explained. Finally, we will use our system on a data set consisting of written documents belonging to the Cultural Heritage field.

Document recommendation is the task of recommending the right documents to archivists or researchers. Several works have been proposed to solve this problem as mentioned in section 2.4.2. Since there is not much work in this field and the ones that do not follow new technologies, none of them will solve the problem. The system we presented while making the document recommendation includes a pipeline that first converts the textual data into a graph structure, then models this graph structure and generates embeddings for the nodes inside the graph. This pipeline can be viewed in Figure 4.1.

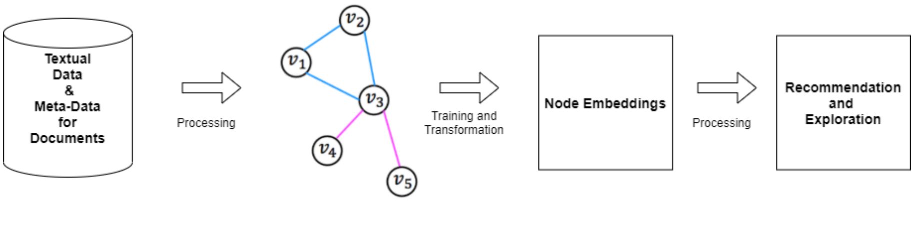

Textual Data & Meta-Data for Documents
Processing
v_2
v_1
v_3
v_4
v_5
Training and Transformation
Node Embeddings
Processing
Recommendation and Exploration

Figure 4.1 System Architecture

4.2 Preliminary - Theory

4.2.1 Graphs and Network Science

Graphs are a popular data structure and a universal model for describing many complex systems in physical, biological (Mashaghi, Ramezanpour & Karimipour, 2004), (Shah, Ashourvan, Mikhail, Pines, Kini, Oechsel, Das, Stein, Shinohara, Bassett & others, 2019), social and information systems (Adalı & Ortega, 2018). The term network is occasionally described as a graph in which attributes correspond to the nodes and edges, and the subject that expresses and investigates the real-world systems as a network is called network science (Barabási, 2013). From the broadest perspective, a graph represents a collection of objects called nodes (vertices), along with a set of interactions called edges between pairs of these objects. Relationships in different domains can be expressed with graphs, for example, illustrated graph in Figure 4.2 refers social network that includes nodes represent individuals and edges indicate that the nodes are connected to their friends.

The most important terminology related to graphs is the following.

- Formally, a graph structure G is expressed in the form of $G = (V, E)$.
- A graph consisting of nonempty set nodes (vertices) $u \in V$ and nonempty set of edges $v \in E$ between these vertices
- Each edge $v \in E$ has a set of one or two vertices associated with it, which are called its endpoints as argued in Gross & Yellen (2003).
- An edge going from node $u \in$ to node $v \in V$ denoted as $(u, v) \in E$.
- If node u is *adjacent* to node v if they are connected by an edge. Two *adjacent* nodes may be called as *neighbors.*
- A graph can be represent as an *adjacency matrix* $\mathbf{A} \in R^{\|V\| \times \|V\|}$ and $\mathbf{A}[u, v]$ $= 1$ if $(u, v) \in E$, 0 elsewhere. Remind that in the case of simple undirected graphs then adjacency matrix will be a symmetric matrix.

Although there are different types of graphs, only *undirected simple graphs* were used in this study. In undirected simple graphs , there is at most one edge between each pair of vertices, no self-loops and where the edges are all undirected. In undirected graphs following propert must be ensured :

- $(u,v) \in E \iff (v,u) \in E$.

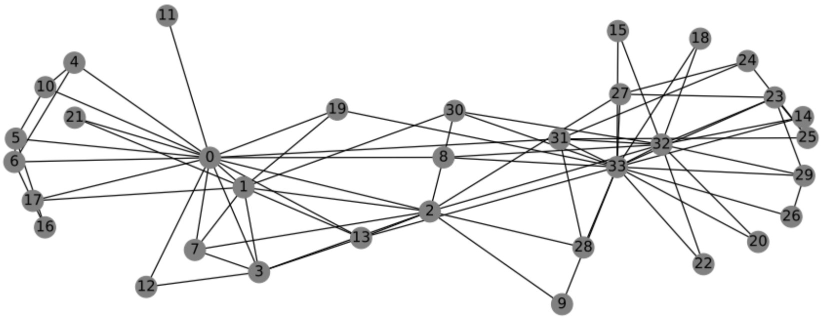


Figure 4.2 The famous Zachary Karate Club Network Zachary (1977) represents the friendship relationships between members of a karate club studied by Wayne W. Zachary from 1970 to 1972 discussed in Hamilton et al. (2017).

4.2.2 Multi-relational Graphs

An important factor in distinguishing graphs is the type of interactions between nodes or edges. For example, graphs can have multiple node types or multiple edge types. Different edge types can specify different relations. In this type of graph, the above-mentioned formulas are expanded to separate edges and nodes according to relation types. At the same time, different adjacency matrices can be created for different relation types.Such graphs called multi-relational, and can be expressed by an adjacency matrix $\mathbf{A} \in R^{\|V\| \times \|R\| \times \|V\|}$, where R is the set of relations (Hamilton et al., 2017).

4.2.3 Heterogeneous Graph & Network

In Heterogeneous graphs there are multiple types of vertices and edges exist. In other words, one can partition the set of nodes U into disjoint sets, such property can be expressed as $V = V_1 \cup V_2 \cup V_3 ... \cup V_n$ where $V_i \cap V_j = \varnothing$ and $\forall i \neq j$. In heterogeneous networks, there are different types of edges between different node types.

Heterogeneous Network : A Heterogeneous Network expressed as a graph in the form of $G = (V, E)$ consisting of a non-empty object set V and a non-empty set of link E. Each node $v \in U$ and each link $e \in E$ associated with a mapping function $\phi(v) \rightarrow T_v$ and $\phi(e) \rightarrow T_e$, where T_v and T_e denotes the type of v and e , $|T_v| + |T_e| > 2$. The network constructed in this study is a special heterogeneous information network graph under the category called the *Multi-partite* graphs , where edges in the graph can only connect vertices that have different types as described in Hamilton et al. (2017).

4.2.4 The Metapath2vec Framework

The following is the procedure to apply the Metapath2vec method on a heterogeneous network.

- Setting & determining meaningful meta-paths.
- Generate a corpus using meta-path-based random walks.
- Incorporate the heterogeneous network structures into skip-gram

4.2.4.1 Metapath Scheme

Meta-Path : Meta-path is described as a path of linking two different object type on network schema. As discussed in Sun & Han (2013), meta-path scheme ρ is expressed as the following :

$$v_1 \xrightarrow{R_1} v_2 \xrightarrow{R_2} .. \xrightarrow{R_{n-1}} v_n \tag{4.1}$$

where $v_n \in V$ denotes the type of nodes and R denotes the set of composite relations along v_1 to v_n . In addition, as stated in Dong et al. (2017) meta-paths are commonly used in a symmetric way , which means first node type v_1 should be the same with the node type last v_n in p.

4.2.4.2 Corpus Generation Using Meta-path-based Random Walks

Proposed Meta-path-based random walk strategy in Dong et al. (2017) states that the semantic relationships between diferent types of nodes can be properly incorporated into skip-gram. Metapath2vec explains the meta-path random walk traversal protocol as given heterogenius Network $G = (V, E, T)$ with $|T_V| > 1$ and meta-path scheme ρ (see section 4.2.4.1) the transition probability at step i is described as follows:

$$p(v^i + 1|v_t^i, \rho) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases} \tag{4.2}$$

where $v_t^i \in V_t$ and $N_{t+1}(v_t^i)$ indicate the V_{t+1} type of neighborhood of node v_t^i.

4.2.4.3 Skip-Gram

In Natural Language Processing, Mikolov et al. (2013) proposed one of the unsupervised learning techniques called Skip-Gram, used to find the most related words for a given word. Word2vec algorithm utilizes the skip-gram method which aims to predict surrounding words in a sentence given a focus word in a sentence and a window with including the surrounding words, resulting in embedding vectors for words. As discussed in the Mikolov et al. (2013) , objective of skip-gram model is the maximizing the average log probability of a sequence of training words $w1, w2, w3, \ldots$ w_T.

$$\arg\max_{\theta} \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, J \neq 0} \log P(w_{t+j}|w_t; \theta) \tag{4.3}$$

, where c is the size of the training context.

Skip-Gram Application on Homogenius Networks

Both DeepWalk and Node2vec algorithms took advantage of random walks and the skip-gram model to learn the representation of a vertice that facilitates the prediction of its structural context—local neighborhoods—in a homogeneous network. Then, objective is became to maximize the network probability in terms of local structures , given a network $G = (V,E)$ as follows :

$$\arg\max_{\theta} \prod_{v \in V} \prod_{c \in N(v)} p(c|v;\theta) \tag{4.4}$$

where $N(v)$ is the neighborhood of node v in the network G and $p(c|v;\theta)$ denotes the conditional probability of having a context node c given a node v.

Skip-Gram Application on Heterogeneous Networks

Finally, Dong et al. (2017) develops the core idea behind Word2vec by applying the Skip-gram algorithm to heterogeneous networks and introduces the heterogeneous skip-gram model to model the heterogeneous neighborhood of a node. As described by Dong et al. (2017)), objective of proposed skip-gram is to maximize the probability of having the heterogeneous context $N_t(v), t \in T_V$ given a node v in heterogenius Network $G = (V,E,T)$ with $|T_V| > 1$ as follows :

$$\arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} \log p(c_t|v;\theta) \tag{4.5}$$

where $N_t(v)$ denotes neighborhood of v's with the t^{th} type of nodes and $p(c_t|v;\theta)$ is commonly described as a softmax function (Dong et al., 2017).

4.2.4.4 Generating Node Embeddings

To give a general summary of Metapath2vec, the generated heterogeneous graph is traversed according to the pre-defined meta path scheme ρ to generate sequences of nodes including a node itself and its neighboring nodes. Such corpus of sequences of nodes is then given as input for the heterogeneous skip-gram model to produces embeddings for each node with the main objective that nodes with similar node neighborhoods should produce similar vector embeddings. The extracted vector embeddings for nodes in a heterogeneous network are in the same vector space, even if they represent different node types.

Formally, Node embeddings are obtained by solving the problem described below,

Given a heterogeneous network G, the task is to learn d - dimensonal latent representations $\boldsymbol{X} \in R^{|V| \times d}$, $d \ll |V|$.

The output of the problem described above is the low-dimensional matrix $\mathbf{X}$, with the v^{th} row-a d-dimensional vector X_v-represents the embedding for node $v \in V$.

4.2.4.5 Top-N Recommendation System

The essential goal in recommendation system is to provide a ranked top-N recommendation list for input queries. In this study, top-n items for particular object is calculated using cosine similarity described as following:

$$cos(\theta) = \frac{X_u \cdot X_v}{\|X_u\| * \|X_v\|}$$

(4.6)

4.3 System Implementation

The general architecture of the system presented throughout the study and the flow of the data used between the systems can be observed in Figure 4.3. In general, the data used first goes through a textual pre-processing stage, then the features that will be used in the future stages are extracted from the data. The extracted features are used both in the construction of heterogeneous graphs and in the implementation of the steps applied under the Metapath2vec technique. Finally, the produced embeddings are used for further recommendation, exploration, and visualization.

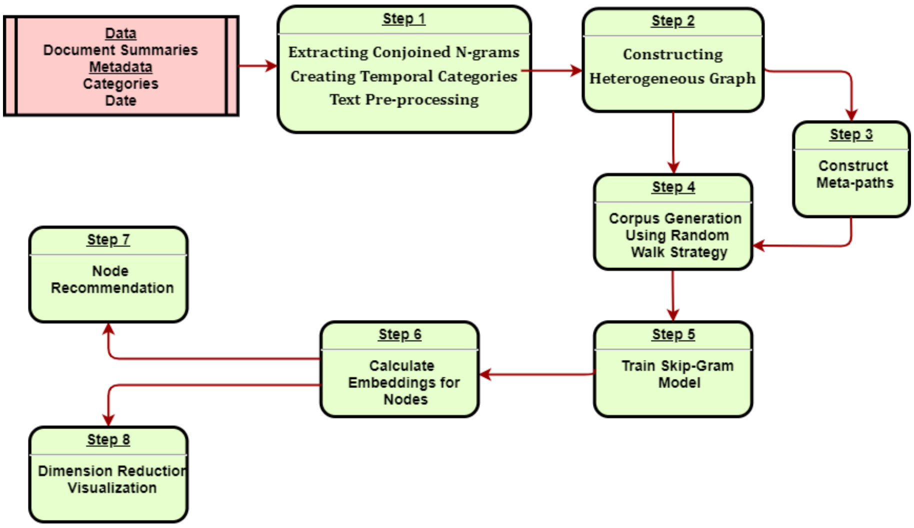


Figure 4.3 Implementation Pipeline and Data Flow

4.3.1 Pre-processing and Feature Extraction

Previous experiments have demonstrated that different pre-processing methods have a significant effect on the performance of many machine learning models. These methods can enhance the model's performance by reducing the dimensionality of data and also reduce the noise in the inputs (Haddi, Liu & Shi, 2013). In this study, common and CH-specific pre-processing steps are applied to the data. The steps are the following:

- Converting all text to lowercase
- Removing all symbols
- Removing all punctuation
- Tokenization using NLTK tokenizer.

4.3.1.1 Temporal Categories

Temporal information in documents means natural language phrases that refer directly to time points or intervals in texts. Depending entirely on the data used, in some cases, this information can be extracted from the texts using Natural Language Processing techniques (Ahn, Adafre & De Rijke, 2005), and in some cases, it has already been extracted. In the data used in this study, temporal information has already been included in the metadata and refers to the date the documents were written. As stated in Section 5.1 that there is also category information in the data used in addition to the time information. There are 14 categories in the collected data, and each document has a category to which it belongs. The category distribution can be observed in Figure 5.3, and the temporal distribution of documents can be observed in Figure 5.4.

Constructing a heterogeneous network is one of the major requirements to take advantage of the Metapath2vec method and the constructed graph in this study described in Section 4.3.3.1. In graph construction, the design of the node types to be used in graph generation is a very critical point. One method to be able to describe the concept of time on graphs was to create different graphs for different time intervals, but with this method, it would not be possible to create representations in the same vector space for nodes in different time zones.

Based on this motivation, we have produced new features that we called "temporal category" as a way of expressing both temporal information and categorical information in the same node type. What is meant by Temporal Category is the name we give to obtain a new attribute by combining the attributes that contain temporal information and also categorical information in the data used. Combining temporal information and categorical would allow, for example, to filter documents on both categorical and temporal properties. To explain the applied technique with an example, the year distribution in Figure 5.4 was divided into n different parts so that each part had numerically close amounts of documents, and these were encoded according to categories (e.g. For finance category, n different new features have been created for the finance category, such as finance_1840_1860, finance_1860_1880 which refers to documents written between 1860 and 1880 and belonging to the finance category).

4.3.1.2 Conjoined N-grams

In computational linguistics and probability, an n-gram refers contiguous sequence of n items from a given sample of text. Such items can be either letter, word, phonemes, syllables, or base pairs according to the application type (Siddharthan, 2002). In this study, by n-grams, we mean words, that is, the smallest element that makes up paragraphs and sentences. Latin numerical prefixes have been used to express n-gram types in the literature, such as an n-gram of size 1 is called as a "*unigram*" , n-gram of size 2 is referred as "*bigram*" , size 3 called as "*trigram*" and n-gram of size 4 is *"four-gram"*, and so on.

The term *conjoined n-gram* also called as *sumgrams* and correspond the most frequent *sumgrams* [1] in the collection. Sumgrams refers to higher-order n-grams (e.g., "world health organization") generated by conjoining lower-order n-grams (e.g., "world health" and "health organization"). In this study, we generate conjoined n-grams different n-gram classes (bigrams, trigrams, k-grams, etc.) as part of the document summaries, instead of limiting the summary to a single n-gram class (e.g., bigrams)

4.3.2 Constructing Heterogeneous Document Network

A constructed heterogeneous document network is a special heterogeneous graph under the category called the *Multi-partite* graphs as mentioned in section 4.2.3. This type of graph can only contain edges that can connect to different node types. The proposed graph consists of three types of nodes and two types of edges, as shown in Figure 4.7. According to the specifications in Section 4.2.3, a heterogeneous network containing 3 different node types was created. The constructed heterogeneous network includes 13547 conjoined n-grams, 7336 documents, 74 temporal categories, with a total of 20957 nodes and 30395 edges, depicted on Figures 4.4,4.5 and 4.6. Detailed Node type descriptions and their corresponding objects can be seen below.

- **Document:** Represents documents in the document collection used as inputs.
- **Conjoined N-Grams :** Represents n-grams extracted from texts as specified in 4.3.1.2.
- **Temporal Categories :** Generated from metadata of documents (see Section 4.3.1.1)

As denoted in Figure 4.7 , three different node types represent documents, sumgrams and temporal categories, respectively. Edge types are the following:

- **Document - Temporal Category** : Edge is constructed if the document is written in the category and time interval represented by the temporal category node.
- **Document - Conjoined N-gram** : Edge is constructed if document contains

[1] https://github.com/oduwsdl/sumgram

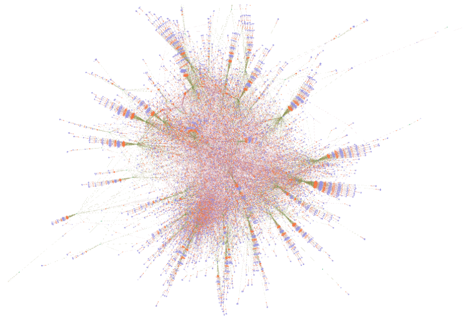


Figure 4.4 Constructed Heterogeneous Document Network, Visualized using Gephi.

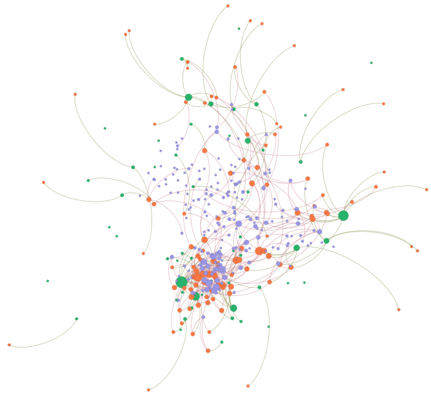


Figure 4.5 Degree Filtered version of proposed Network, Green nodes represents Temporal Categories, Red represents Conjoined N-grams and Blue nodes represents Documents. Visualized using Gephi.

particular n-gram.

Figure 4.8 includes an example of how the proposed heterogeneous graph is constructed utilizing the previously mentioned node types and relations coupling them.

4.3.3 Document Representation

We used the Metapath2vec method in our system to find the node representations corresponding to the documents in the data we utilize. However, to compare our system, we also obtained vector representations for nodes with different methods including TF-IDF, DOC2VEC, and BERT.

4.3.3.1 Document Representation Using Metapath2vec

In order to obtain embeddings for different node types on the constructed heterogenious network, the methodology presented by Dong et al. (2017) and explained in the Section 4.2.4 was followed and implemented using Networkx (Hagberg, Swart & S Chult, 2008) , Gensim (Rehurek & Sojka, 2011) , and Stellar Graph (Data61,

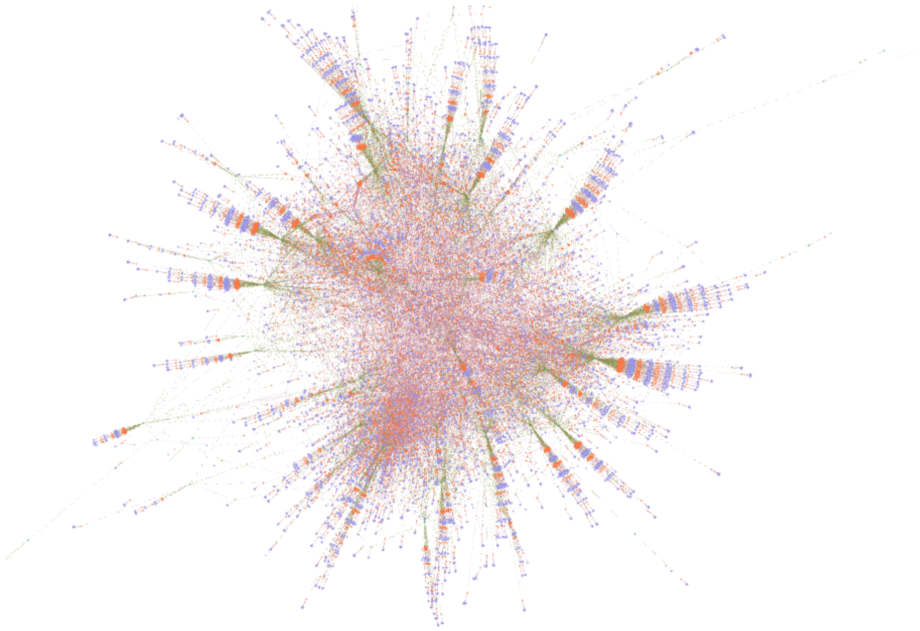


Figure 4.6 Proposed Heterogeneous Document Network, Visualized using Gephi.

2018) libraries.

Setting Meta-Path Scheme : The meta-path that we have defined in our system by following the Equation 4.1 is as follows.

$$v_{N-gram} \xrightarrow{R_1} v_{Document} \xrightarrow{R_2} v_{Temporalcategory} \xrightarrow{R_2} v_{Document} \xrightarrow{R_1} v_{N-gram}$$

Setting Parameters for Random Walks and Skip Gram :

For skip-gram and random walk strategy ,the hyper-parameters used are listed below.

- The number of walks per root node w:2,
- The walk length l :30,
- vector dimension d: 128,
- neighborhood size k: 10,
- size of negative samples: 5,

4.3.3.2 Document Representation Using TF-IDF

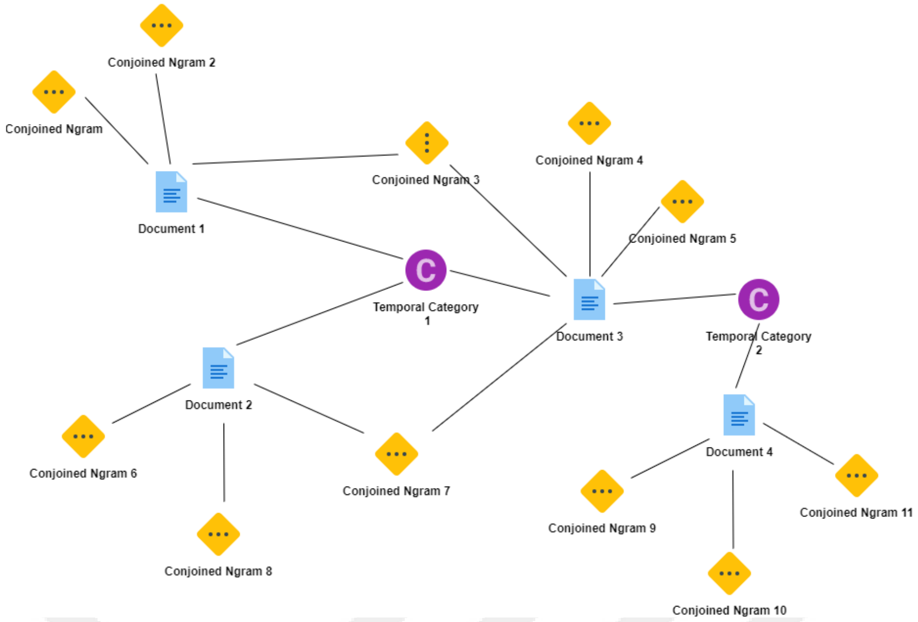


Figure 4.7 Proposed Heterogeneous Information Network Design for Document - Conjoined Ngram - Temporal Category Relationship

TF-IDF refers term frequency-inverse document frequency, which is a popular method the natural language processing. TF-IDF accounts for the relative frequency of words in a particular document through an inverse proportion of the word along with all documents in a corpus. *TF* represents term frequency of term i in a document j while *IDF* refers inverse document frequency of term i (Arroyo-Fernández, Méndez-Cruz, Sierra, Torres-Moreno & Sidorov, 2019) (Schmidt, 2019). TF-IDF score of a word in documents expressed by Manning & Raghavan (2008) as

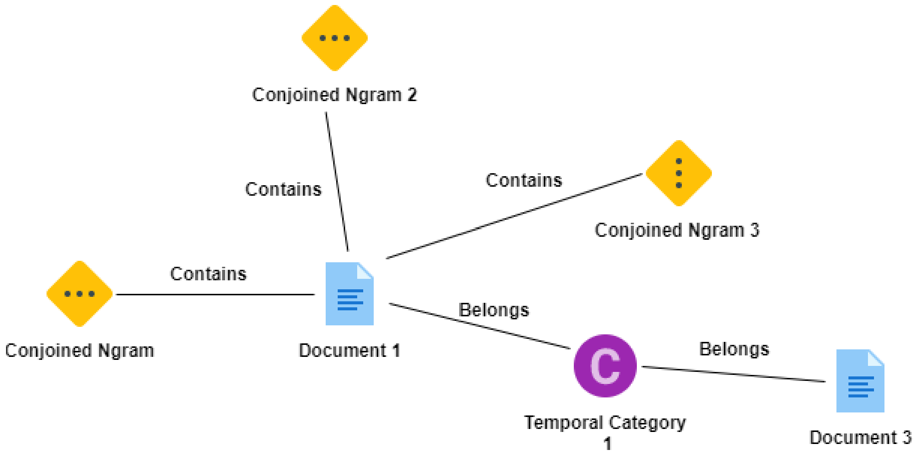


Figure 4.8 Example of Illustration of Proposed Graph and Relations Between Nodes

following.

$$\mathrm{tf}(t,d) \equiv \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

(4.7)

where $f_{t,d}$ is the raw count of a term in a document.

$$\mathrm{idf}(t,D) \equiv \log \frac{N}{|d \in D : t\ ind|} \tag{4.8}$$

where N represents total number of document in the corpus $N = |D|$ and $|d \in D : t\ ind|$ denotes the number of documents where term t appears.

$$\mathrm{tfidf}(t,d,D) \equiv tf(t,d) \times idf(t,D) \tag{4.9}$$

To learn vector representation of document embeddings we followed the same procedure stated in the Dai, Olah & Le (2015) and treat the classical bag-of-words model where each word is represented as a one-hot vector weighted by TF-IDF. Finally, the document is represented by a constructed vector.

4.3.3.3 Document Representation Using Doc2vec

Le & Mikolov (2014) proposed Doc2vec which is an extension for word2vec (Mikolov et al., 2013). The authors aim to extend the learning of embeddings from words to word sequences (e.g. paragraphs) by describing a paragraph vector, an unsupervised learning method that learns vector embeddings for variable-length pieces of texts including sentences and documents. Doc2vec proposes Distributed Bag-of-Word (see Figure 4.9) in the same way as skip-gram 4.2.4.3, except that input is altered by token that represents the particular document. To learn vector representation of document embeddings , we followed the procedures presented in Le & Mikolov (2014) and Lau & Baldwin (2016).

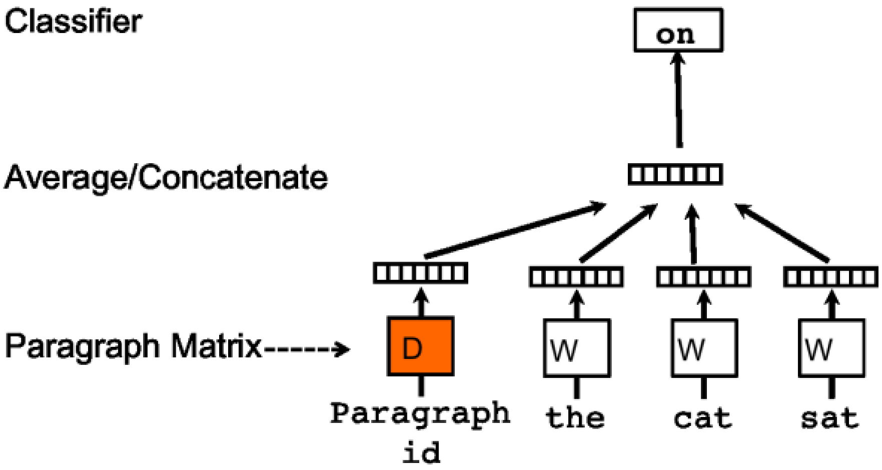


Figure 4.9 An Architecture for learning paragraph vector. Distributed Bag of words Retrieved from Le & Mikolov (2014). Important difference with traditional skip-gram is the additional paragraph token that is mapped to a vector via matrix D.

4.3.3.4 Document Representation Using BERT

BERT (Devlin, Chang, Lee & Toutanova, 2018), which stands for a Bidirectional Encoder Representations from Transformers (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser & Polosukhin, 2017), is a deep bidirectional encoder network based on a purely transformer architecture. Bert is capable of the capture the bidirectional representations of texts by jointly conditioning on both left and right context in all layers. Furthermore, a pre-trained BERT model can be applied a broad range of tasks by fine-tuning with just one additional output layer. We used the publicly available BERTurk-Base cased model [2] that pre-trained on unlabeled Turkish corpus which has a size of 35 GB text. BERTurk model consists of 12 transformer layers, 768 hidden states size. We fine-tuned the model by adding one extra dropout and a fully connected layer that has 768 input and 2 output units at the end of the network and jointly trained it with the BERTurk model.

We applied the sentence-transformation framework proposed by Reimers & Gurevych (2019) on the top of the pre-trained BERTurk-Base cased model to obtain document embeddings. The proposed sentence-transformation technique enables the BERT model to compute dense vector representations for sentences, paragraphs, and images.

[2] https://github.com/stefan-it/turkish-bert

4.3.4 Document Recommendation

To perform document recommendation, first document embedding was calculated for a query document, then the closest Top - K document embedding was calculated using cosine similarity and suggested. We have shown in previous chapters and sections that the Metapath2vec algorithm is used to calculate these embeddings. To compare the proposed system with baselines, we explained how to calculate embedding for documents using TF-IDF, Doc2vec, and BERT in the previous sections.

4.3.4.1 Top-K Document Recommendation Using Metapath2vec

The description of our proposed algorithm that makes top - k recommendations for documents using the Metapath2vec model can be seen in Algorithm 1. To summarize, a heterogeneous graph structure was constructed using the features we extracted using the summaries of the documents and the metadata of the documents. Then, we generated a corpus of node sequences using the random walk strategy over this graph structure and obtained the vector representations for the nodes using the heterogeneous skip-gram algorithm. Finally, we used cosine similarity to determine the distance (similarity) between the query node and the remaining nodes others and suggest the top k similar nodes in the type of query node.

Algorithm 1 Metapath2vec Document Recommendation

```
Input: The training documents $d^T = \{n, t, s\}$ , sumgram $n$ , temporal category $t$, document summary $s$ , a meta-path scheme $\rho$ , walks per node $w$ , walk length $l$ , embedding dimension $d$, neighborhood size $k$, query node $q_n$ , top k element $k$
Output: First k element in $S \in R^{\|V\| \times d}$

Initiliaze X , empty list $S$ ;
Construct Heterogeneous Graph $G = (V, E, T)$ using $d^T$;
$\mathbf{X}$ = Metapath2vec($G$,$\rho$,$v$,$l$);
$S$ = Recommend($q_n$,$\mathbf{X}$);
return $S$;

Metapath2vec($G$,$\rho$,$v$,$l$)
for $i = 1$ →w do
    for $v$ ∈V do
        update $\mathbf{X}$ according to the steps in 4.2.4 ;
    end for
end for
return $\mathbf{X} \in R^{\|V\| \times d}$;

Recommend($q_n$,$\mathbf{X}$)
for  each $x \in \mathbf{X}$ do
    calculate cosine similarity $\theta$ between $x$ and $q_n$ according to the Equation 4.6;
    insert ($x$,$\theta$) to $S$;
end for
sort $S$ in increasing $\theta$;
return $S$;
```

Chapter 5. Results

In this chapter, we discuss a systematic evaluation process performed to assess the performance of our proposed system. First, we describe a dataset we have developed our system on. After that we introduce quality and objectivity metrics, motivating its adoption and giving some intuition for how it works.

5.1 Dataset Description

As a result of our conversations with professional archivists in the Salt team, we decided to use Waqfs of Crete which is an archive consisting of official records of Muslim inhabitants of Crete who moved to Turkey during the 1920s due to the population exchange between Turkey and Greece. Crete is the largest and most populous of the Greek islands, and the fifth-largest island in the Mediterranean Sea (see Figure 5.2), after Sicily, Sardinia, Cyprus, and Corsica. It bounds the southern border of the Aegean Sea. Crete rests approximately 160 km south of the Greek mainland. Crete was under Ottoman rule until 1898, and there was an independent Crete State until 1908; later on, Crete became part of Greece. In the 19th century, there were large Muslim and Christian groups with occasional uprisings from both parties (Kostopoulou, 2016).

Documents spanning the period from 1825 to 1928 in Ottoman Turkish and Greek provide an opportunity to examine the multi-layered social structure on the island, especially from a cultural and economic perspective. The metadata of Waqfs of Crete provided by a SALT Research team comprises a specialized library and an archive of physical and digital sources and documents on visual practices, the built environment, social life, and economic history in Turkey. The metadata contains information for approximately 10 thousand documents and includes the summary of those documents, the year they were published, the location, the language used, and the documents' picture. An example document and its metadata are depicted in Figure 5.1 . More information about the data can be found below.

The metadata of 10145 documents including

- *Image of Document* : Pictures of documents in the collection which originally written in Ottoman. An example document picture can be seen in Figure 5.1.
- *Summary* : Text that summarizing the event depicted by pictorially available documents. These texts were written by archive professionals. The average number of tokens in these summaries is calculated as 30.
- *Date* : The date the documents were written. Year distribution can be shown in Figure 5.4.
- *Language* : The language of the documents is Modern Turkish but most of the words are of Ottoman origin, which is difficult for a normal person to read.
- *Category* : Documents are organized according to the 14 categories they belong to by archive professionals, these categories include Aid, Complaints, Education, Families, Finance, Inheritance, Military Service, Miscellaneous, Orphans, Personnel, Population Exchange, Properties, Request for Protection and Verdicts. Category distribution can be shown in Figure 5.3.

5.2 Testing NLP Enhanced Visual Exploration Tool

5.2.1 Use Case: Waqfs of Crete

In the proposed visual interface in Section 3, each of the nodes on the map in the interface represents the documents in the collection of Waqfs of Crete. These documents may consist of deeds of properties owned by people, the lawsuits they have filed, the permits received from the municipality, and various other official documents. In the application designed for the user to find the relationship between documents, the keyword, and individual search components narrow the document pool to be researched on. In this way, the user/archivists may find documents that are roughly related to each other very quickly. Later, the user can browse each

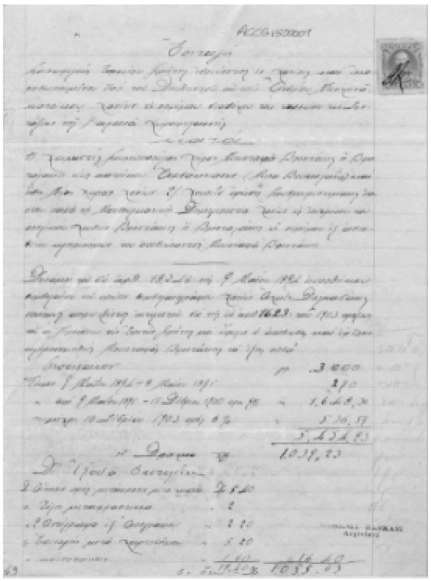


Figure 5.1 Example of a single document and its metadata, *Summary :* The Director of the Public Benefit Treasury of Crete, which succeeded the treasury of pensions of the gendarmerie, Andreou Moshona, acknowledges to the heirs of the Deceased Mustafa Vrodaki or Vrodalaki that unless the debt of deceased Mustafa Vrodaki of a total sum of 1.055,63 drachmas is paid off in half a month, the deceased's real estate property comprised of fields in Village Ardaktiana and of their house inside the village will be sold at public auction. *Category :* Inheritance , *Date :* 1911-12-15

document on its network and view documents that have a specific relationship to each other. As an example of this specific relationship, user can easily observe the actions that a specific person has made in his entire life, such as which specific people he/she communicates with, individuals' family, get married, socially interact with their environment, purchasing, die, and leave a legacy. Besides, with the temporal expressions, the user can follow these interactions like a story and do this for all the people who lived in the 19^{th} century on Crete's island.

5.3 Experimental Design for Document Recommendation

5.3.1 Manual Annotation Setup

Figure 5.2 Crete Island

To evaluate our document recommendation system trained on the Waqfs of Crete dataset, we conducted an experiment with 5 archive professionals and 5 non-professional reviewers on the Waqfs of Crete dataset. The rules of the organized experiment can be seen below.

- 2 query documents were randomly selected for each of the 5 most popular categories. For the query documents that we selected randomly, we suggested the 10 documents that our proposed metapath2vec (see 4.3) model recommends as the most relevant to the query document.
- Each participant was given a total of 10 documents, including 2 documents from each category, and 50 documents suggested by our model. Throughout the experiment, we asked them to annotate the documents suggested by the model as "Relevant", "Irrelevant" or “Language is not clear”.
- In order to measure the agreement among the participants, we enforce 50% (5 out of 10 documents) overlap between query documents that are given to each annotator.
- Documents were selected from the categories of "Family", "Property", "Finance", "Personal", and "Orphans". These are the 5 most common categories in the data.

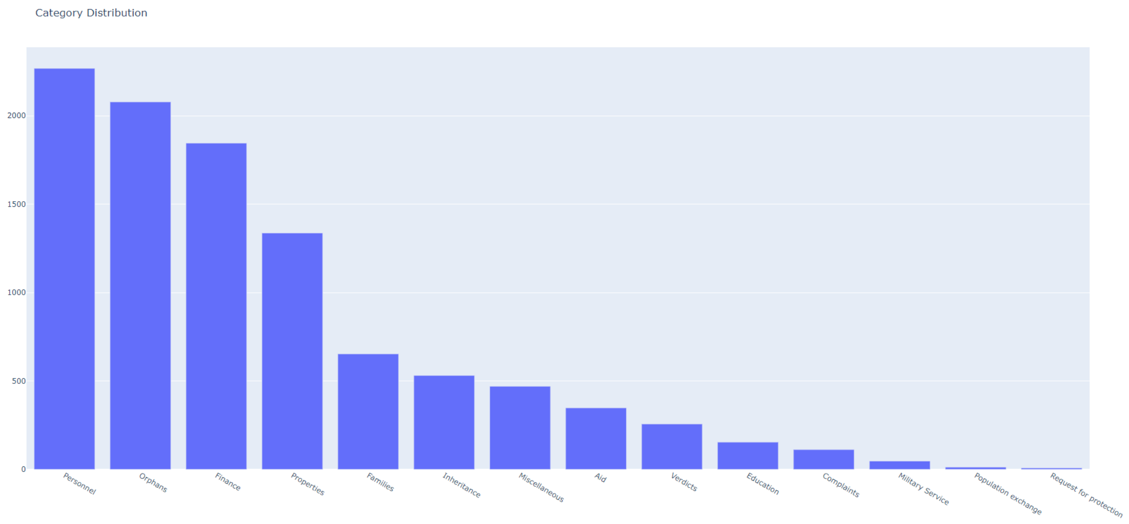


Figure 5.3 Category Distribution of Waqfs of Crete Dataset

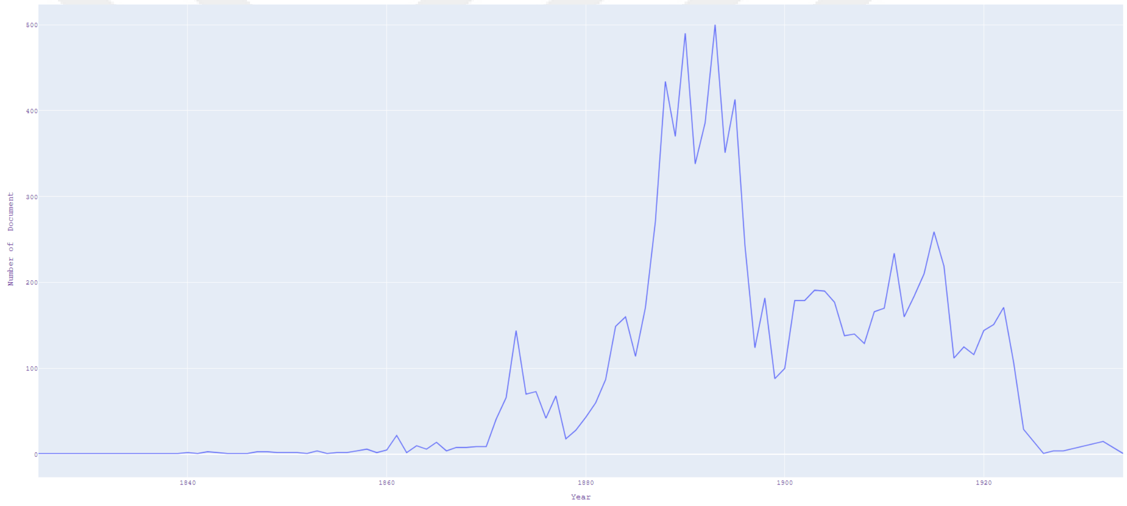


Figure 5.4 Year Distribution of Waqfs of Crete Dataset

5.3.2 Inter-Rater Reliability (IRR)

In statistics, inter-rater reliability (also called inter-annotator-agreement) is the degree of agreement among different raters as described by Saal, Downey & Lahey (1980). It is a score of how much homogeneity or consensus exists in the ratings given by various judges. Different machine learning tasks have always needed labeled data that is frequently annotated by humans. We conducted two different Inter-Rater Reliability (IRR) techniques to measure how well multiple annotators can make the same annotation decision for a certain category on different objects. The annotation scheme used in this study was to annotate the recommended document for a query document as *Relevant*, *Not Relevant* or *Language is not clear* . As we have stated in the Experiment specifications (Section 5.3.1), %50 percent of the documents we provide to users to measure the annotator agreement are mutual to everyone participating in the experiment. We used Cohen's Kappa and Percent Agreement to measure Agreement. The methodologies underlying the techniques and the results can be seen in the following section.

5.3.2.1 Percent Agreement :

The percent agreement refers to the joint probability of agreement and is the simplest and the least IRR measure. It is predicted as the percentage of the time the annotators agree in a nominal or categorical rating system.It does not consider the fact that agreement may happen only based on chance (Uebersax, 1987). To calculate the measure of percent agreement between two raters, we followed the following procedures :

- Calculate the number of ratings in agreement.
- Calculate the total number of ratings.
- Divide the total by the number of ratings in agreement.

5.3.2.2 Cohen's Kappa Coefficient K :

Cohen's kappa coefficient is a statistic which measures the inter-rater agreement for categorical items (McHugh, 2012). Cohen's kappa is thought to be a more reliable measure than simple percent agreement calculation, as considering the possibility of the agreement occurring by chance. K measures the agreement between two raters who each classify N objects into C mutually exclusive categories. The definition of Cohen's Kappa is the following.

$$K \equiv \frac{p_o - p_e}{1 - p - e} = \frac{1 - p_o}{1 - p_e} \tag{5.1}$$

where p_o is the relative observed agreement among raters , and p_e is the hypothetical probability of chance agreement.

5.3.2.3 Evaluation of Inter-Rater Reliability

Cohen suggested the Kappa result be interpreted as follows:

- values 0 as indicating no agreement
- 0.01–0.20 as none to slight
- 0.41– 0.60 as moderate
- 0.61–0.80 as substantial
- 0.81–1.00 as almost perfect agreement (McHugh, 2012).

Our experiment yielded an average pairwise agreement of 75.2% and moderate inter-annotator agreement (Cohen's = 0.49) between all 10 annotator while average pairwise agreement of 82% and substantial inter-annotator agreement (Cohen's = 0.61) between 5 professional annotator. The results of pairwise agreement scores among all participants are shown in Figure 5.6, while the scores among only professional participants are shown in Figure 5.5. Similarly, the results of Cohen's Kappa scores among all participants are shown in Figure 5.8, while the scores among only professional participants are shown in Figure 5.7.

5.3.3 Experiment Evaluation & Results

For comparison, cosine similarity scores between the selected embedding of documents and the embedding of recommended documents were recalculated using 3 comparison models including TF-IDF, Doc2vec, and BERT. A detailed description of the recommendations we made using these models is available in Chapter 4.3.3.2, 4.3.3.3 and 4.3.3.4. In summary, the similarity score between the documents proposed by our model (10 documents per query document) for the query documents and the query documents (10 in total) was re-calculated with these 3 models. In order to compare the proposed recommendations in the most accurate and objective way, we have followed the following procedures.

1.1 Participants had annotated the recommended documents for query documents as Relevant, Not Relevant or Language is Not Clear. An important point here is that the documents recommended for the query document are the Top - N recommendation produced by our proposed model (see Chapter 4.3.3.1, that is, the 10 documents with the highest similarity for the query document, as stated in 4.3.4.

1.2 We produced the similarity score between the Query documents and the documents our model recommends, with 3 extra models in the same way. In other words, embeddings are reobtained between the recommended documents and the query documents with 3 extra different models.

1.3 The similarity scores have been recalculated between document embeddings produced by the models including TF-IDF, Doc2vec, and BERT.

1.4 We evaluate the results of experiment using the Mann-Whitney U test (Mann & Whitney, 1947) (see Algorithm 5.2) according to the answers given by the participants in the experiment. Mann-Whitney U test is a statistical test that can be used to characterize the degree of separation between two frequency distributions. Using this test, we wanted to prove that there is a difference between similarity score distributions of differently labeled documents.

Mann-Whitney U Test : Mann-Whitney U test is a non-parametric statistical technique that analyzes the differences between the medians of two sets. It tests the null hypothesis that is equally likely that a randomly selected value from one sample will be less than or greater than a randomly selected value from the second sample (Mann & Whitney, 1947).

Formally, Mann-Whitney U statistic is defined as:

$$U = \sum_{i=1}^{n}\sum_{j=1}^{m} S(X_i, Y_j), \qquad S(X_i, Y_j) = \begin{cases} 1 & Y < X \\ \frac{1}{2} & Y = X \\ 0 & Y > X \end{cases}$$

(5.2)

We applied the Mann-Whitney U test and compared the test statistics to measure and compare which model yielded more significant test statistics. According to our hypothesis, documents annotated as *Relevant* should yield higher similarity scores than documents marked as *Not Relevant*. In this way, we applied the hypothesis tests in two different ways. First, we calculated the test statistics between the similarity scores of the documents marked "relevant" and "not relevant" for each user. This resulted in 10 different test statistics for 10 participants per model and the distribution of user test statistics can be seen in Figure 5.9.In addition, the methodology we followed while obtaining test statistics for users is illustrated on Algorithm 2. The kernel density plot of our proposed model and the models we compared, drawn from the test statistics on a random participant, illustrated in Figure 5.11.

Second, we analyzed U-statistics by document category, that is, we extracted test statistics between the similarity scores of annotated documents for each category. In this way, 5 test statistics for 5 categories per model were extracted and their distribution is shown in Figure 5.10.In addition, the methodology we followed while obtaining test statistics for categories is illustrated on Algorithm 3. The kernel density plot of our proposed model and the models we compared, drawn from the test statistics on a *Families* category, illustrated in Figure 5.12. According to the results in Table the model we presented showed better results than TF-IDF and Doc2vec models in both types of experiments and is slightly behind BERT.

According to the results in Table 5.1, the model we proposed yields better results than TF-IDF and Doc2vec models in both types of experiments and is slightly behind BERT.

Algorithm 2 U-Statistic-Users

```
Input: model $m$ , participant list $\mathbf{P}$
Output: non-empty list of U-Statistics $U_{users}$
initialize empty list $U_{users}$;
for each $p_i \in \mathbf{P}$ do
    Retrieve $l_{relevant}$ and $l_{irrelevant}$[1] for $p_i$ from $m$;
    Calculate $U_{pi} \equiv S(\mathrm{l}_{relevant}, l_{irrelevant})$;
    Insert $U_{pi}$ into $U_{users}$;
end for
return $U_{users}$;
```

Algorithm 3 U-Statistic-Categories

```
Input: model $m$ , category list $\mathbf{C}$
Output: non-empty list of U-Statistics $U_{users}$
initialize empty list $U_{categories}$;
for each $c_i \in \mathbf{C}$ do
    Retrieve $l_{relevant}$ and $l_{irrelevant}$[2] for $c_i$ from $m$;
    Calculate $U_{ci} \equiv S(\mathrm{l}_{relevant}, l_{irrelevant})$;
    Insert $U_{ci}$ into $U_{categories}$;
end for
return $U_{categories}$;
```

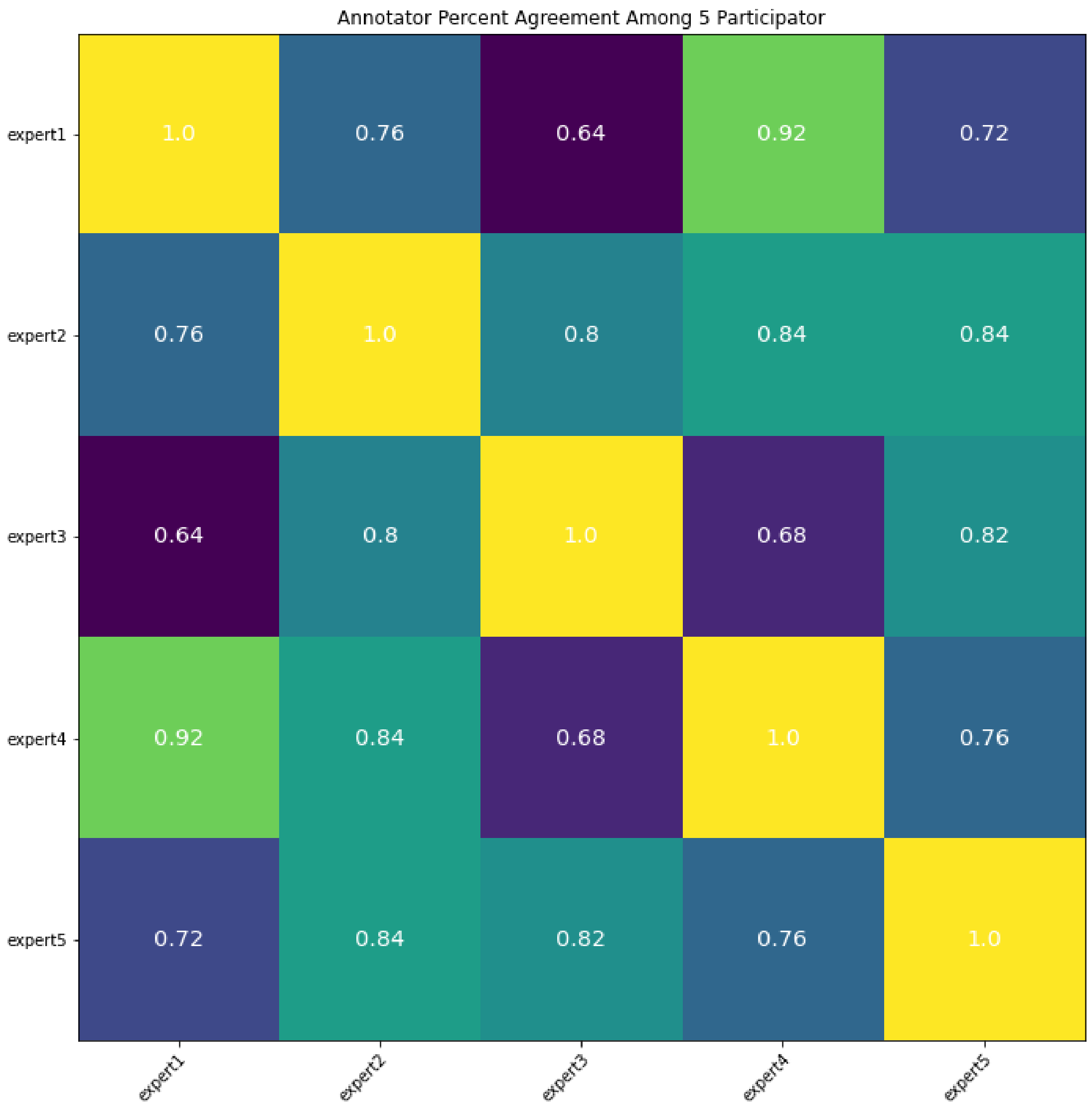
[1] $l_{relevant}$ and $l_{irrelevant}$ represents lists of similarity scores for relevant and irrelevant documents annotated

Model	Experiment Type	Mean U-Statistics
Our Model	Among Participants	936.7
TF-IDF	Among Participants	371.2
Doc2vec	Among Participants	863.8
Bert	Among Participants	987.8
Our Model	Among Categories	921.1
TF-IDF	Among Categories	389.1
Doc2vec	Among Categories	798.9
Bert	Among Categories	1035.5

Table 5.1 Evaluation of Models

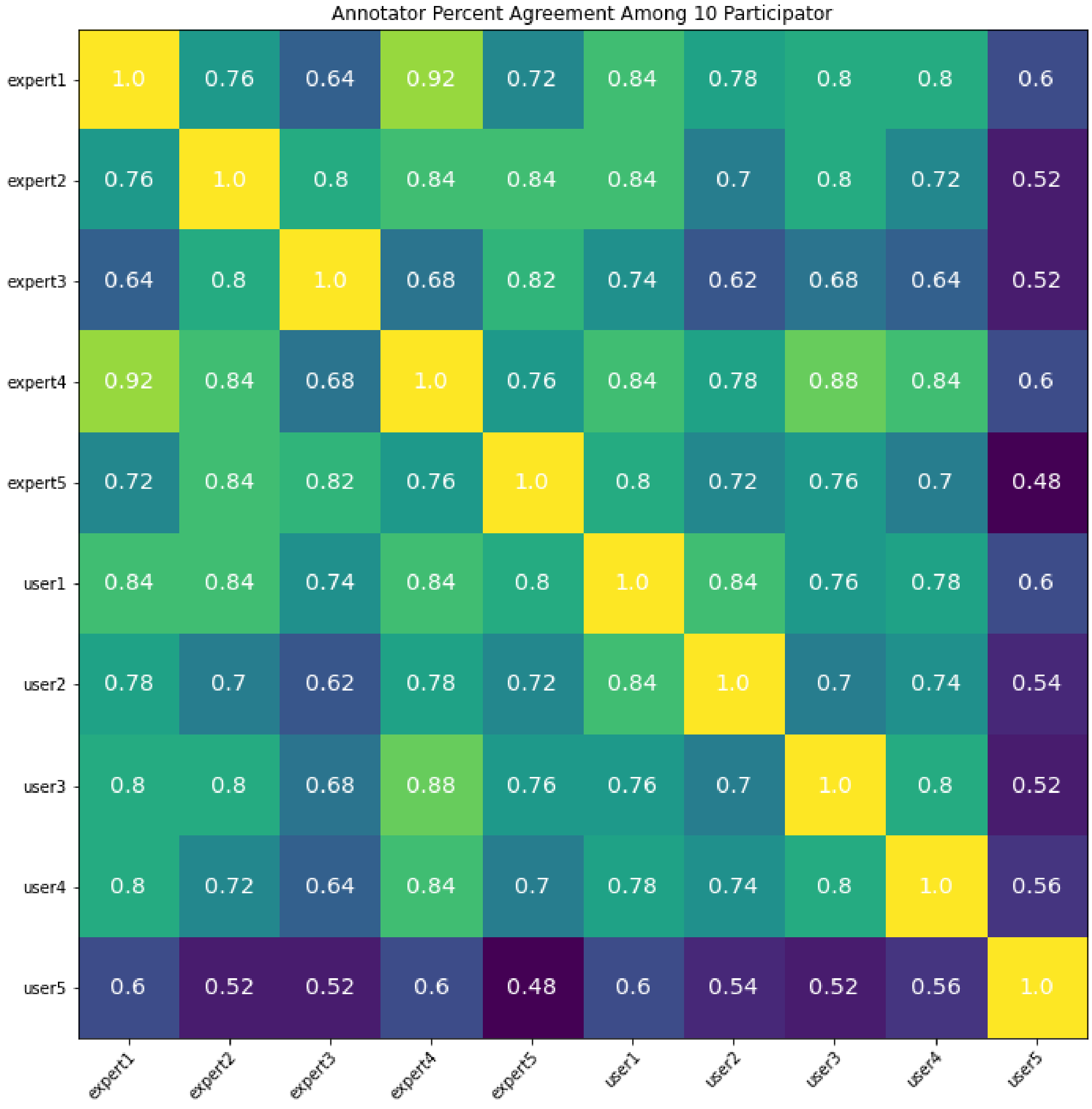
by p_i .

[2] $l_{relevant}$ and $l_{irrelevant}$ represents lists of similarity scores for relevant and irrelevant annotated documents under c_i .



	expert1	expert2	expert3	expert4	expert5
expert1	1.0	0.76	0.64	0.92	0.72
expert2	0.76	1.0	0.8	0.84	0.84
expert3	0.64	0.8	1.0	0.68	0.82
expert4	0.92	0.84	0.68	1.0	0.76
expert5	0.72	0.84	0.82	0.76	1.0

Figure 5.5 Percent Agreement Among 5 Professional Participator



	expert1	expert2	expert3	expert4	expert5	user1	user2	user3	user4	user5
expert1	1.0	0.76	0.64	0.92	0.72	0.84	0.78	0.8	0.8	0.6
expert2	0.76	1.0	0.8	0.84	0.84	0.84	0.7	0.8	0.72	0.52
expert3	0.64	0.8	1.0	0.68	0.82	0.74	0.62	0.68	0.64	0.52
expert4	0.92	0.84	0.68	1.0	0.76	0.84	0.78	0.88	0.84	0.6
expert5	0.72	0.84	0.82	0.76	1.0	0.8	0.72	0.76	0.7	0.48
user1	0.84	0.84	0.74	0.84	0.8	1.0	0.84	0.76	0.78	0.6
user2	0.78	0.7	0.62	0.78	0.72	0.84	1.0	0.7	0.74	0.54
user3	0.8	0.8	0.68	0.88	0.76	0.76	0.7	1.0	0.8	0.52
user4	0.8	0.72	0.64	0.84	0.7	0.78	0.74	0.8	1.0	0.56
user5	0.6	0.52	0.52	0.6	0.48	0.6	0.54	0.52	0.56	1.0

Figure 5.6 Percent Agreement Among All Participators

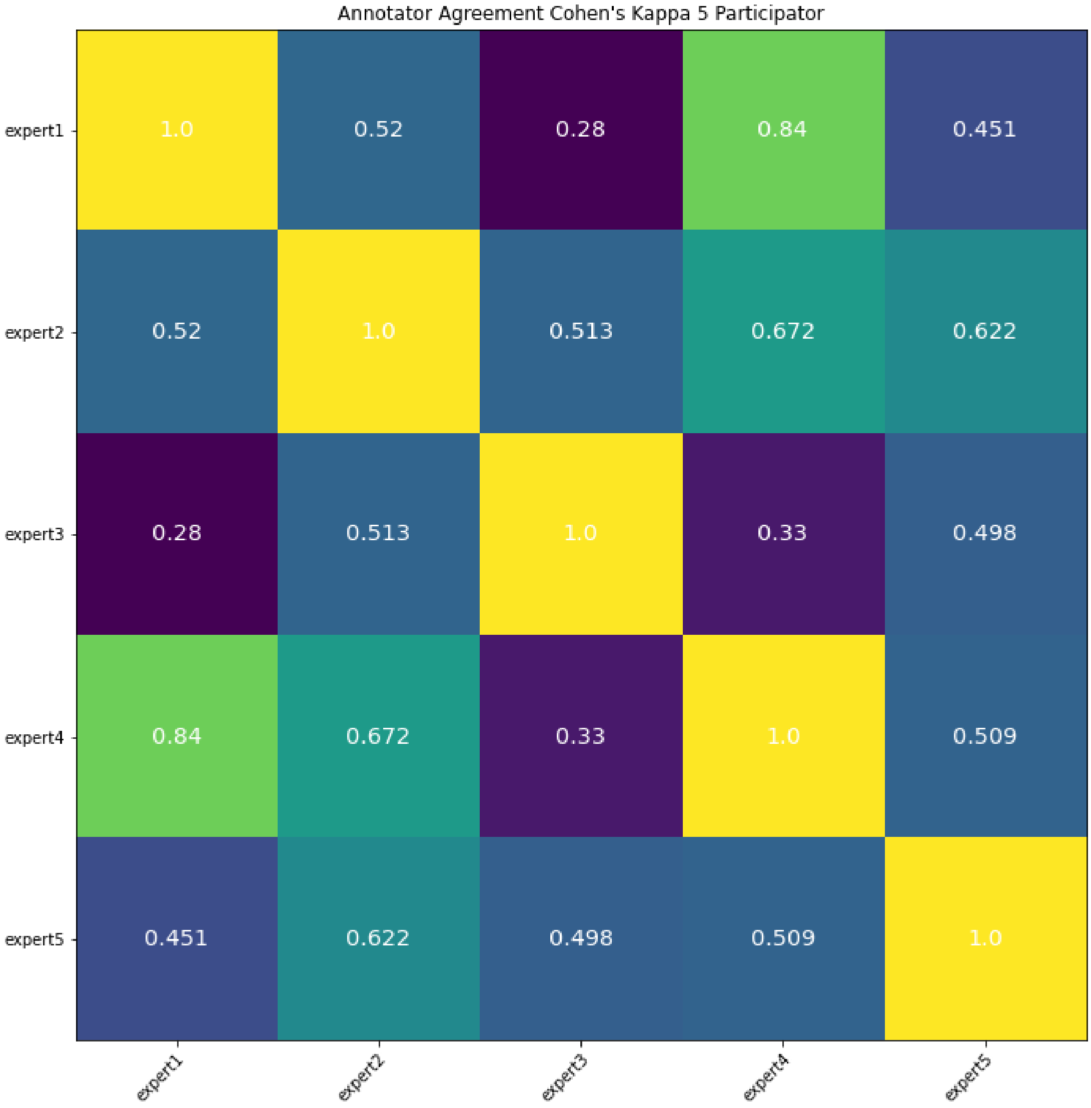
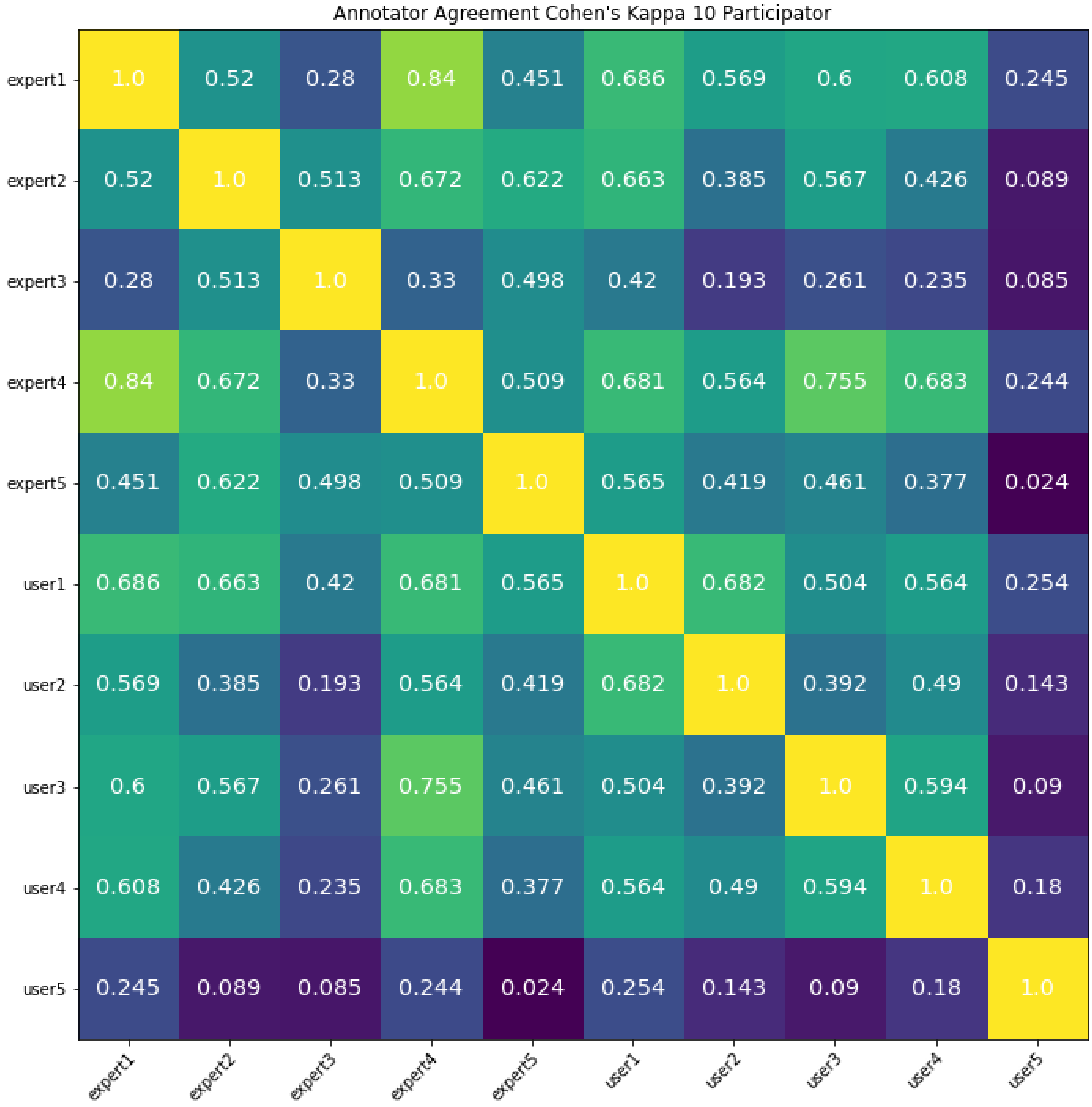


Figure 5.7 Cohen's Kappa Among 5 Professional Participator



	expert1	expert2	expert3	expert4	expert5	user1	user2	user3	user4	user5
expert1	1.0	0.52	0.28	0.84	0.451	0.686	0.569	0.6	0.608	0.245
expert2	0.52	1.0	0.513	0.672	0.622	0.663	0.385	0.567	0.426	0.089
expert3	0.28	0.513	1.0	0.33	0.498	0.42	0.193	0.261	0.235	0.085
expert4	0.84	0.672	0.33	1.0	0.509	0.681	0.564	0.755	0.683	0.244
expert5	0.451	0.622	0.498	0.509	1.0	0.565	0.419	0.461	0.377	0.024
user1	0.686	0.663	0.42	0.681	0.565	1.0	0.682	0.504	0.564	0.254
user2	0.569	0.385	0.193	0.564	0.419	0.682	1.0	0.392	0.49	0.143
user3	0.6	0.567	0.261	0.755	0.461	0.504	0.392	1.0	0.594	0.09
user4	0.608	0.426	0.235	0.683	0.377	0.564	0.49	0.594	1.0	0.18
user5	0.245	0.089	0.085	0.244	0.024	0.254	0.143	0.09	0.18	1.0

Figure 5.8 Cohen's Kappa Among All Participators

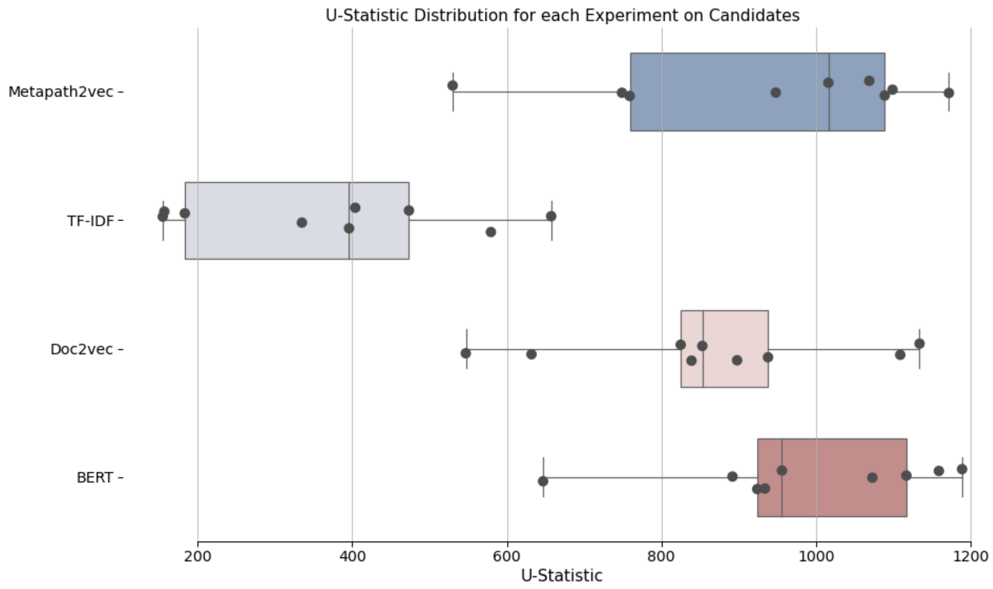


Figure 5.9 U-Statistic Distribution for each Experiment on Candidates

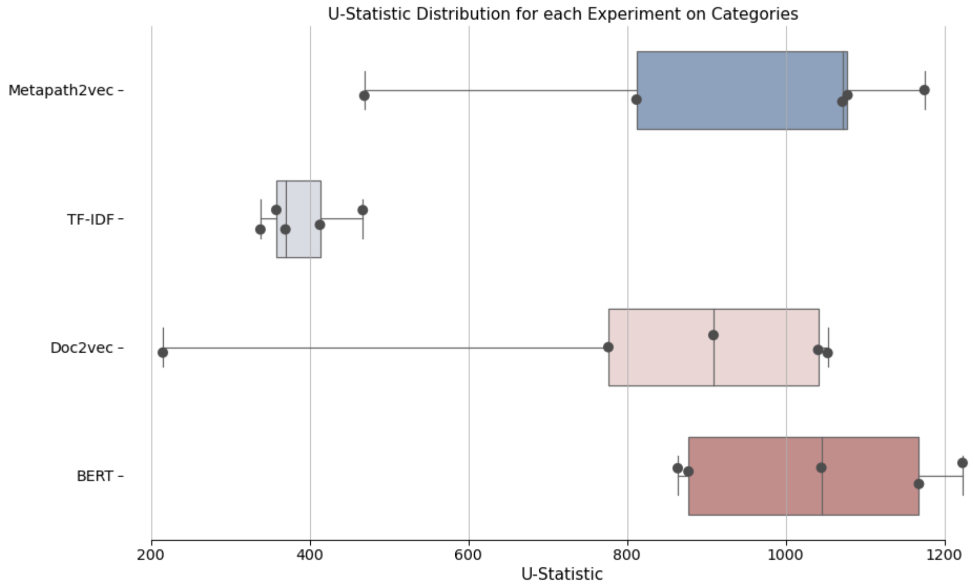


Figure 5.10 U-Statistic Distribution for each Experiment on Categories

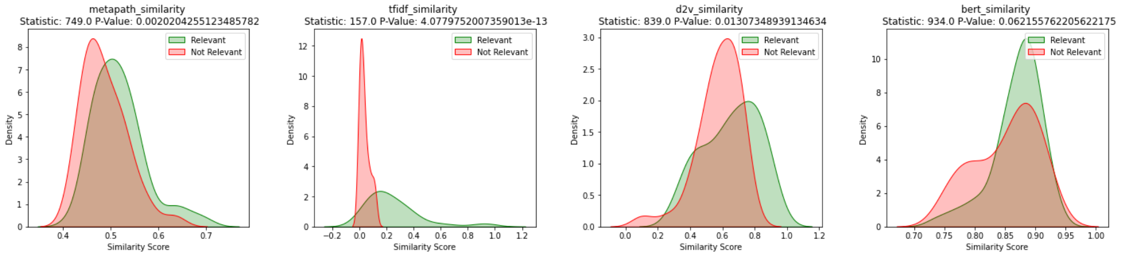


Figure 5.11 A kernel density estimate for test statistics distribution on users for each model.

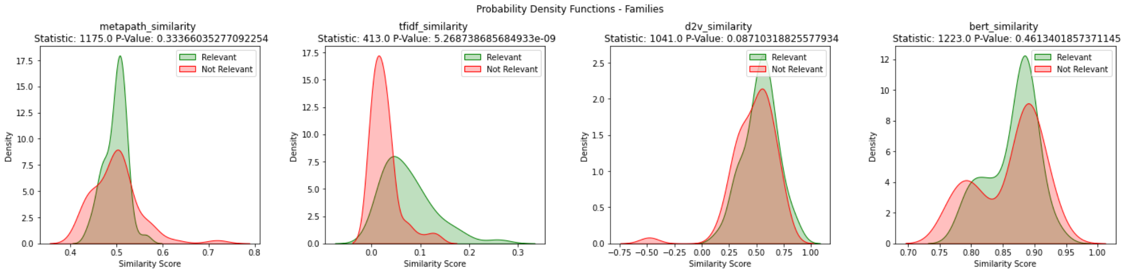


Figure 5.12 A kernel density estimate for test statistics distribution on categories for each model.

Chapter 6. Conclusion & Future Works

In this thesis, two different systems were developed and tested on a collection of documents in the field of cultural heritage. To access such a collection, we have collaborated with archive professionals from a cultural institution, SALT which focused on public service producing research-based exhibitions, publications, and digitization projects in Turkey. The first of the two systems presented throughout the thesis is a visual analytics tool, and the second is a document recommendation system designed using the texts and metadata of the documents in the entire document collection.

The proposed visual exploratory tool is demonstrated to uncover hidden information and stories underlying documents, extracting the key attributes within the documents, and establishing a network between documents using various NLP and visualization methods. Thanks to the developed tool we tested on the Archive of Waqfs of Crete, users may observe people's specific actions in the documents and get a chance to witness a person living in the 1900s life. The application was only tested and designed on the Waqfs of Crete archive, but it needs to be tested on many more collections in different domains. In this way, the relationships between documents in different domains are more noticeable. Moreover, user studies with detailed questionnaires must be carried out on a group of archivists, and received responses must be analyzed to evaluate further the performance of the tool. And finally, the designed tool must be compared with the existing solutions in the field so that a better evaluation of the tool can be obtained.

The document recommendation system we have introduced constructs heterogeneous document networks using novel feature extraction procedures, such as using conjoined n-grams and creating temporal categories to describe the data in the best way. In the meantime, we only benefited from the textual summaries of the documents and the metadata of the documents. Then, using the graph structure created, the system adopts heterogeneous skip-gram and random walk strategies, which are the techniques used by the recently proposed Metapath2vec algorithm, generates embeddings for documents, and uses these embeddings in the recommendation system.

The extracted embeddings were especially used to design the document suggestion system, but visual results were also obtained by visualizing these embeddings.

We designed an experiment with 5 archive professionals and 5 normal users to test the performance of the recommendation system we propose. During the experiment, we supply query documents and recommended the top 10 results suggested by our proposed model for query documents. Finally, asked them to annotate these suggested documents as *Relevant* , *Not Relevant* and *Language is Not Clear* . It took a long time to organize and perform the experiments, as the language of the documents was difficult for a normal person to understand. To compare the results, we used the Mann-Whitney U-test (Mann & Whitney, 1947), a non-parametric hypothesis test, and our hypothesis was that documents marked as "Relevant" had higher similarity scores than documents marked "Not Relevant" for a query document. In this framework, we compared the models with each other according to their significant test statistics according to the experimental results. As a result of the experiments, the model we proposed gave better results than the alternative document embedding methods, these are TF-IDF and Doc2vec, but they gave a very similar result with the BERT. In a conclusion, considering that BERT is a model that is trained with very large amounts of data and is too parametric and very complex, our result is reasonable.

As future work, the different n-grams we extract in the created network actually correspond to the people, places, and locations living in that period due to the nature of the data. We do not have a system that can classify what kind of entity the n-grams we extract correspond to yet, but we can make inferences with certain heuristic-based methods (e.g. identify n-grams those containing Turkish proper names and classified them as n-grams expressing persons). We are planning to focus on this part and classify the entity type represented by n-grams, because with such an approach, we will be able to better understand and capture the social fabric under the island of Crete, and we aim to examine entities referring people, locations and events and the naturally occurring networks among these entities. On the other hand, we are working on methods that will combine the graph and visualization parts in the most accurate way. We will visualize the extracted node embeddings in 2 or 3 dimensional planes and integrate them on the application we have presented.

BIBLIOGRAPHY

Adalı, T. & Ortega, A. (2018). Applications of graph theory [scanning the issue]. *Proceedings of the IEEE*, *106*(5), 784–786.

Ahn, D., Adafre, S. F., & De Rijke, M. (2005). Extracting temporal information from open domain text: A comparative exploration. In *Information Retrieval Workshop*, (pp.3̃).

Aho, A. V. & Corasick, M. J. (1975). Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, *18*(6), 333–340.

Arroyo-Fernández, I., Méndez-Cruz, C.-F., Sierra, G., Torres-Moreno, J.-M., & Sidorov, G. (2019). Unsupervised sentence representations as word information series: Revisiting tf–idf. *Computer Speech & Language*, *56*, 107–129.

Aviles Collao, J., Diaz-Kommonen, L., Kaipainen, M., & Pietarila, J. (2003). Soft ontologies and similarity cluster tools to facilitate exploration and discovery of cultural heritage resources. *IEEE Computer Society Digital Library. Proc. DEXA*, 1–5.

Barabási, A.-L. (2013). Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *371*(1987), 20120375.

Belkin, M. & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips*, volume 14, (pp. 585–591).

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Bhagat, S., Cormode, G., & Muthukrishnan, S. (2011). Node classification in social networks. In *Social network data analytics* (pp. 115–148). Springer.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Brochier, R., Guille, A., & Velcin, J. (2019). Global vectors for node representations. In *The World Wide Web Conference*, (pp. 2587–2593).

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, *34*(4), 18–42.

Budzik, J. & Hammond, K. J. (2000). User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th international conference on intelligent user interfaces*, (pp. 44–51).

Burnard, L. & Short, H. (1994). An arts and humanities data service. *Report of a Feasability Study commissioned by the Information Systems Sub-Committee of the Joint Information Systems Committee of the Higher Education Funding Councils*.

Chen, H., Perozzi, B., Hu, Y., & Skiena, S. (2018). Harp: Hierarchical representation learning for networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chen, H. M., Zou, H., & Scott, A. L. (2016). Improving the analysis and retrieval of digital collections: a topic-based visualization model. *Journal of Information Technology Management*, *27*(2), 82–92.

Cheng, K. S., Wang, Z., Huang, P.-C., Chundi, P., & Song, M. (2020). Topexplorer: Tool support for extracting and visualizing topic models in bioengineering text corpora. In *2020 IEEE International Conference on Electro Information Technology (EIT)*, (pp. 334–343). IEEE.

Chorowski, J., Weiss, R. J., Bengio, S., & van den Oord, A. (2019). Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, *27*(12), 2041–2053.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, *12*(ARTICLE), 2493–2537.

Dahl, G., Ranzato, M., Mohamed, A.-r., & Hinton, G. E. (2010). Phone recognition with the mean-covariance restricted boltzmann machine. *Advances in neural information processing systems*, *23*, 469–477.

Dai, A. M., Olah, C., & Le, Q. V. (2015). Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.

Data61, C. (2018). Stellargraph machine learning library. `https://github.com/stellargraph/stellargraph`.

De Boom, C., Van Canneyt, S., Demeester, T., & Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, *80*, 150–156.

Deng, L., Seltzer, M. L., Yu, D., Acero, A., Mohamed, A.-r., & Hinton, G. (2010). Binary coding of speech spectrograms using a deep auto-encoder. In *Eleventh Annual Conference of the International Speech Communication Association*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, Y., Chawla, N. V., & Swami, A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, (pp. 135–144).

Dou, J., Qin, J., Jin, Z., & Li, Z. (2018). Knowledge graph based on domain ontology and natural language processing technology for chinese intangible cultural heritage. *Journal of Visual Languages & Computing*, *48*, 19–28.

Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology*, *38*(1), 188–230.

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Gross, J. L. & Yellen, J. (2003). *Handbook of graph theory*. CRC press.

Grover, A. & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 855–864).

Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., & He, X. (2010). Document recommendation in social tagging services. In *Proceedings of the 19th international conference on World wide web*, (pp. 391–400).

Gupta, S. & Varma, V. (2017). Scientific article recommendation by using distributed representations of text and graph. In *Proceedings of the 26th inter-*

national conference on world wide web companion, (pp. 1267–1268).

Gwinn, N. E. & Rinaldo, C. (2009). The biodiversity heritage library: sharing biodiversity literature with the world. *IFLA journal*, *35*(1), 25–34.

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*, 26–32.

Hagberg, A., Swart, P., & S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Hampson, C., Agosti, M., Orio, N., Bailey, E., Lawless, S., Conlan, O., & Wade, V. (2012). The cultura project: supporting next generation interaction with digital cultural heritage collections. In *Euro-Mediterranean Conference*, (pp. 668–675). Springer.

Hart, P. E. & Graham, J. (1997). Query-free information retrieval. *IEEE Expert*, *12*(5), 32–37.

Hoff, P. D., Raftery, A. E., & Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the american Statistical association*, *97*(460), 1090–1098.

Kong, X., Mao, M., Wang, W., Liu, J., & Xu, B. (2018). Voprec: Vector representation learning of papers with text information and structural identity for recommendation. *IEEE Transactions on Emerging Topics in Computing*.

Kostopoulou, E. (2016). The island that wasn't: Autonomous crete (1898–1912) and experiments of federalization. *Journal of Balkan and Near Eastern Studies*, *18*(6), 550–566.

Kristensson, P. O., Arnell, O., Björk, A., Dahlbäck, N., Pennerup, J., Prytz, E., Wikman, J., & Åström, N. (2008). Infotouch: an explorative multi-touch visualization interface for tagged photo collections. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, (pp. 491–494).

Lau, J. H. & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Le, Q. & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, (pp. 1188–1196).

Lee, K. C. & Kwon, S. (2008). Online shopping recommendation mechanism and its influence on consumer decisions and behaviors: A causal map approach. *Expert Systems with Applications*, *35*(4), 1567–1574.

Li, X., Lin, S., Yan, S., & Xu, D. (2008). Discriminant locally linear embedding with high-order tensor data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(2), 342–352.

Liben-Nowell, D. & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, *58*(7), 1019–1031.

Liu, Z., Lin, Y., & Sun, M. (2020). *Representation Learning for Natural Language Processing*. Springer Nature.

Maltz, D. & Ehrlich, K. (1995). Pointing the way: Active collaborative filtering. In *Proceedings of the SIGCHI conference on Human factors in computing sys-*

tems, (pp. 202–209).

Mann, H. & Whitney, D. (1947). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Ann. Math. Stat*, *18*(1), 50–60.

Manning, C. D. & Raghavan, P. (2008). and schutze, h.[2008] introduction to information retrieval.

Mashaghi, A. R., Ramezanpour, A., & Karimipour, V. (2004). Investigation of a protein complex network. *The European Physical Journal B-Condensed Matter and Complex Systems*, *41*(1), 113–121.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica*, *22*(3), 276–282.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, (pp. 3111–3119).

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5115–5124).

Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Nagori, R. & Aghila, G. (2011). Lda based integrated document recommendation model for e-learning systems. In *2011 international conference on emerging trends in networks and computer communications (ETNCC)*, (pp. 230–233). IEEE.

Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, *104*(1), 11–33.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 701–710).

Perozzi, B., Kulkarni, V., Chen, H., & Skiena, S. (2017). Don't walk, skip! online learning of multi-scale network embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, (pp. 258–265).

Petridis, P., Pletinckx, D., Mania, K., & White, M. (2006). The epoch multimodal interface for interacting with digital heritage artefacts. In *International Conference on Virtual Systems and Multimedia*, (pp. 408–417). Springer.

Rehurek, R. & Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, *3*(2).

Reimers, N. & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Reimers, N. & Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1–35). Springer.

Saal, F. E., Downey, R. G., & Lahey, M. A. (1980). Rating the ratings: Assessing the psychometric quality of rating data. *Psychological bulletin*, *88*(2), 413.

Sak, H., Güngör, T., & Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International Conference on Natural Language Processing*, (pp. 417–427). Springer.

Salisu, S. M., Mayr, E., Filipov, V. A., Leite, R. A., Miksch, S., & Windhager, F. (2019). Shapes of time: Visualizing set changes over time in cultural heritage collections. In *EuroVis (Posters)*, (pp. 45–47).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, (pp. 285–295).

Schmidt, C. W. (2019). Improving a tf-idf weighted document vector embedding. *arXiv preprint arXiv:1902.09875*.

Schweter, S. (2020). Berturk - bert models for turkish.

Serdar, A. & Tutumlu, R. (2018). Building a digital ottoman/turkish serial novel archive. *AIUCD 2018*, 33.

Shah, P., Ashourvan, A., Mikhail, F., Pines, A., Kini, L., Oechsel, K., Das, S. R., Stein, J. M., Shinohara, R. T., Bassett, D. S., et al. (2019). Characterizing the role of the structural connectome in seizure dynamics. *Brain*, *142*(7), 1955–1972.

Shaparenko, B. & Joachims, T. (2009). Identifying the original contribution of a document via language modeling. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, (pp. 350–365). Springer.

Siddharthan, A. (2002). Christopher d. manning and hinrich schutze. foundations of statistical natural language processing. mit press, 2000. isbn 0-262-13360-1. 620 pp. 64.95/£ 44.95 (cloth). *Natural Language Engineering*, *8*(1), 91.

Spangler, S., Kreulen, J. T., & Lessler, J. (2002). Mindmap: Utilizing multiple taxonomies and visualization to understand a document collection. In *Proceedings of the 35th annual Hawaii international conference on system sciences*, (pp. 1170–1179). IEEE.

Sun, Y. & Han, J. (2013). Meta-path-based search and mining in heterogeneous information networks. *Tsinghua Science and Technology*, *18*(4), 329–338.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, (pp. 1067–1077).

Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*.

Uebersax, J. S. (1987). Diversity of decision-making models and the measurement of interrater agreement. *Psychological bulletin*, *101*(1), 140.

Van Den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Neural Information Processing Systems Conference (NIPS 2013)*, volume 26. Neural Information Processing Systems Foundation (NIPS).

Van Meteren, R. & Van Someren, M. (2000). Using content-based filtering for recommendation. In *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, volume 30, (pp. 47–56).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, (pp. 5998–6008).

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., & Borgwardt, K. M. (2010). Graph kernels. *Journal of Machine Learning Research*, *11*, 1201–1242.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. (2020). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*.

Wang, Z., Zhu, W., Cui, P., Sun, L., & Yang, S. (2013). Social media recommendation. In *Social media retrieval* (pp. 23–42). Springer.

Weng, S.-S. & Chang, H.-L. (2008). Using ontology network analysis for research document recommendation. *Expert Systems with Applications*, *34*(3), 1857–1869.

Yang, C., Liu, Z., Zhao, D., Sun, M., & Chang, E. (2015). Network representation learning with rich text information. In *Twenty-fourth international joint conference on artificial intelligence*.

Yu, C.-C. & Chang, H.-p. (2009). Personalized location-based recommendation services for tour planning in mobile tourism applications. In *International Conference on Electronic Commerce and Web Technologies*, (pp. 38–49). Springer.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, *33*(4), 452–473.

Zhang, Y., Ai, Q., Chen, X., & Croft, W. B. (2017). Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, (pp. 1449–1458).

Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., & Li, Z. (2018). Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, (pp. 167–176).

APPENDIX A

Visualization of Node Embedding

In this Appendix, we present the visualizations of vector embeddings that we extracted from the constructed heterogeneous graph including node types as document - conjoined n-gram - temporal categories. Embedding vectors are reduced to 2 dimensions with the T-SNE algorithm. In Figure A.2, nodes are colored according to the node types, and among these node types, those that represent real individuals are shown as a separate node type. To identify n-grams that reflect individuals, we do not have a system that can classify what kind of entity the n-grams we extract correspond to yet, but we can make inferences with certain heuristic-based methods. For example, among the sumgrams (conjoined-n-grams) we extracted, we identified those containing Turkish proper names and classified them as n-grams expressing persons. 8792 individuals have been identified using the above heuristic. This ratio accounts for 27% of the n-grams we extract.

In Figure A.1, we can observe the interaction and clustering of documents written at different times and belonging to different categories. Many clusters occur mainly because n-grams that characterize certain events are commonly mentioned in the documentation.

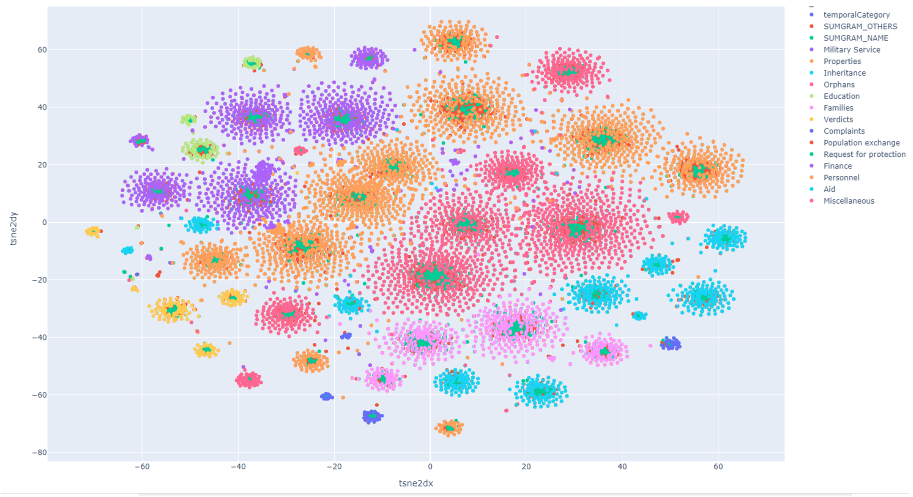


Figure A.1 TSNE Visualization of node embeddings, colored by document categories.

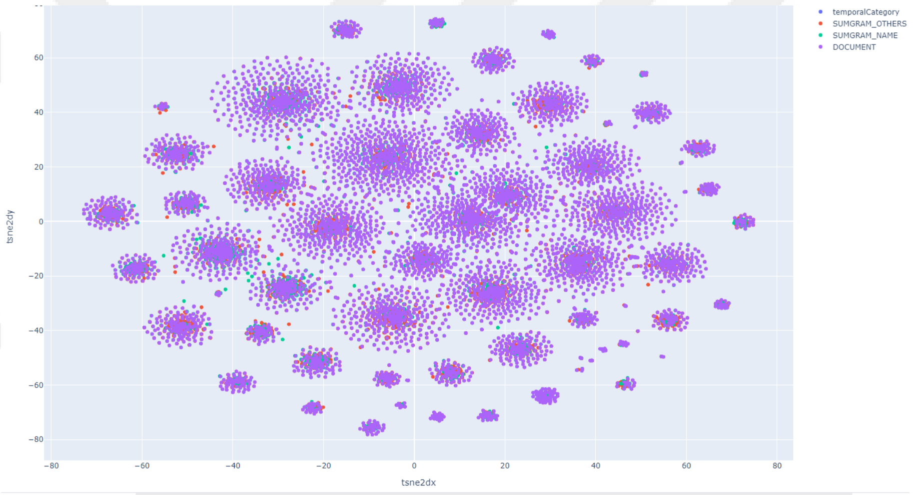


Figure A.2 TSNE Visualization of node embeddings, colored by node type.