Question Answering with Deep Neural Networks for Semi-Structured Heterogeneous Genealogical Knowledge Graphs

Omri Suissa[a], Maayan Zhitomirsky-Geffet[a] and Avshalom Elmalech[a]
[a]*Department of Information Science, Bar Ilan University, omrivm@gmail.com, Israel*

Editor(s): Mehwish Alam, FIZ Karlsruhe - Leibniz Institute for Information Infrastructure, Germany; Victor de Boer, Vrije Universiteit Amsterdam, the Netherlands; Enrico Daga, The Open University, United Kingdom; Marieke van Erp, KNAW Humanities Cluster, the Netherlands; Eero Hyvönen, University of Helsinki, Aalto University, Finland; Albert Meroño Peñuela, Vrije Universiteit Amsterdam, the Netherlands; Harald Sack, FIZ Karlsruhe - Leibniz Institute for Information Infrastructure, Germany
Solicited review(s): Ricardo Usbeck, University of Hamburg, Country; Isaiah Onando Mulang, IBM, Kenya; One anonymous reviewer

Abstract. With the rising popularity of user-generated genealogical family trees, new genealogical information systems have been developed. State-of-the-art natural question answering algorithms use deep neural network (DNN) architecture based on self-attention networks. However, some of these models use sequence-based inputs and are not suitable to work with graph-based structure, while graph-based DNN models rely on high levels of comprehensiveness of knowledge graphs that is nonexistent in the genealogical domain. Moreover, these supervised DNN models require training datasets that are absent in the genealogical domain. This study proposes an end-to-end approach for question answering using genealogical family trees by: 1) representing genealogical data as knowledge graphs, 2) converting them to texts, 3) combining them with unstructured texts, and 4) training a transformer-based question answering model. To evaluate the need for a dedicated approach, a comparison between the fine-tuned model (Uncle-BERT) trained on the auto-generated genealogical dataset and state-of-the-art question-answering models was performed. The findings indicate that there are significant differences between answering genealogical questions and open-domain questions. Moreover, the proposed methodology reduces complexity while increasing accuracy and may have practical implications for genealogical research and real-world projects, making genealogical data accessible to experts as well as the general public.

Keywords: Question answering, Genealogy, Neural Networks, Knowledge graph, Natural Language Processing, Transformers, Cultural heritage

1. Introduction

The popularity of "personal heritage", user-generated genealogical family tree creation, has increased in recent years, driven by new digital services, such as online family tree sharing sites, family tree creation software, and even self-service DNA analysis by companies like Ancestry and My Heritage. These genealogical information systems allow users worldwide to create, upload and share their family tree in a semi-structured graph format named GEDCOM (GEnealogical Data COMmunication)[1]. Most genealogical information systems also provide natural search capabilities (a search engine) to find relatives and related family trees. While the user interface [4, 10, 56, 75, 103] and user interactions [45, 69] with genealogical information systems are well researched, to the best of our knowledge, there is no research on natural question-answering in the genealogical domain for genealogical information systems.

[1] https://www.gedcom.org/

As humans, we are accustomed to asking questions and receiving answers from others. However, the standard search engines and information retrieval (IR) systems require users to find answers from a list of documents. For example, for the question “How many children does Kate Kaufman have?", the system will retrieve a list of documents containing the words "children" and "Kate Kaufman". Unlike search engines and IR systems, natural question answering algorithms aim to provide precise answers to specified questions [47]. Thus, if a user is searching a genealogical database for the family tree of Kate Kaufman[2], a built-in question answering system will not return a list of possible matches but will provide a short and precise answer to various natural language questions. For instance, for a question such as "Where was Kate's father born?", a genealogical question answering system will return the answer "Hesse, Germany". Genealogical centers and museums seek to create a unique and personal experience for visitors using chatbots [73] and even holographic projections of private or famous people [78]. Hence, one practical implication of such a genealogical question answering system can be posing natural questions to a museum holographic character, or even a holographic restoration of a person from a family tree. Imagine walking into a genealogical center and talking to your great-grandmother, asking her questions about your family history and heritage. The underlying technology for such a conversation (inter alia) is based on the ability to answer natural questions on the GEDCOM data of genealogical family trees. The current state-of-the-art method for solving such a task is based on deep neural networks (DNN).

DNN models for open-domain natural question answering achieved high accuracy in multiple studies [15, 102, 116, 118, 119, 120, 127]. Training DNN models for question answering requires a golden standard dataset constructed from questions, answers, and corresponding texts from which these answers can be extracted. An extensive golden standard dataset for the natural question answering task widely used for training such models is Stanford Question Answering Dataset (SQuAD) [90, 91]. However, in the field of genealogy, there are no standard training datasets of questions and answers similar to SQuAD.

Generating a genealogical training dataset for question answering DNN is challenging, since genealogical data constitutes a semi-structured heterogeneous graph. It contains a mix of a structured graph and unstructured texts with multiple nodes and edge types, where nodes may include structured data on a specific person node (e.g., person's birthplace), structured data on a specific family node (e.g., marriage date), relations between nodes, and unstructured text sequences (e.g., bio notes of a person). Such a mix of structured heterogeneous graph data and unstructured text sequences is not the type of input that state-of-the-art models, like BERT [22] and other sequence-based DNN models, are designed to work with.

Therefore, the main objective of the proposed study is to design and empirically validate an end-to-end pipeline and a novel methodology for question-answering DNN using graph-based genealogical family trees combined with unstructured texts.

The research questions addressed in this study are:

1. What is the effect of the training corpus domain (i.e., open-domain vs. genealogical data) and the consanguinity scope on the accuracy of neural network models in the genealogical question answering task?
2. How to traverse a genealogical data graph while preserving the meaning of the genealogical relationships and family roles?
3. What is the effect of the question type on the DNN models' accuracy in the genealogical question answering task?

The main contributions of the study are:

1. A new automated method for question answering dataset generation derived from family tree data, based on the knowledge graph representation of genealogical data and its automatic conversion into a free text;
2. A new graph traversal method for genealogical data;
3. A fine-tuned question answering DNN model for the genealogical domain, Uncle-BERT, based on BERT[3] [22] that outperforms state-of-the-art DNN models (trained for answering open-

[2]https://dbs.anumuseum.org.il/skn/en/c6/e2216499 5/Personalities/Kaufman_Kate

[3] https://huggingface.co/bert-base-uncased

domain questions) for various question types.

2. Related work

This section covers related work in the fields relevant to this research: genealogical family trees, neural network architecture, and question answering using neural networks.

2.1. Genealogical family trees

Genealogical family trees have become popular in recent years. Both non-profit organizations and commercial companies allow users worldwide to upload and update their family tree online. For example, commercial enterprises like Ancestry and My Heritage collect over 100 million[4] and 48 million[5] family trees, respectively; FamilySearch is the largest non-profit online collection of family trees with over a billion[6] unique individuals worldwide. Family trees can be created from various sources, such as family trees uploaded by private users (UGC) [6], clinical reports and DNA records [20, 105], biographical register [64], and even books [27]. Family tree records contain valuable information about individuals and their genealogical relationships, information that is useful for historical research and preservation [46], population and migration research [84], and even medical research [124, 126]. The user-generated content family trees phenomena, also called "personal heritage", combines the study of the history of one's ancestors with local and social history [6]. Figure 1 illustrates the degrees of relationships between two people in the genealogical domain [12].

[4] https://support.ancestry.com/s/article/Searching-Public-Family-Trees

[5] https://www.myheritage.co.il/about-myheritage/

[6] https://www.familysearch.org/en/about

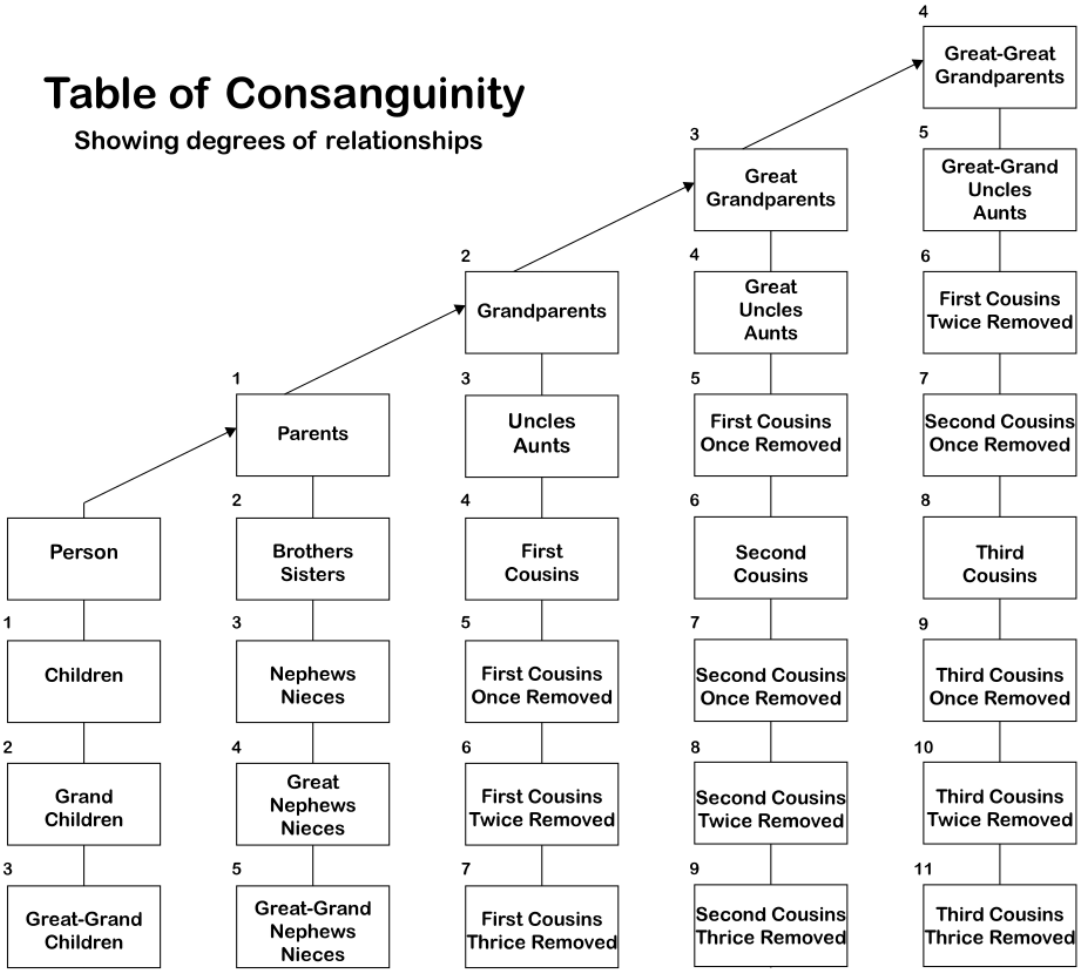

Fig. 1. Relation degrees in genealogy[7].

[7] Figure created by WClarke (https://commons.wikimedia.org/wiki/User:WClarke) based on original by User:Sg647112c - Own work. From Wikipedia: https://en.wikipedia.org/wiki/Consanguinity. CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

2.1.1. The GEDCOM genealogical data standard

The de facto standard in the field of genealogical family trees is the GEDCOM format [36, 56]. The standard developed by The Church of Jesus Christ of Latter-day Saints in 1984, and the latest released version (5.5.1) that was drafted in 1999 and fully released in 2019, still dominates the market [42]. Other standards have been suggested as replacements, but none were extensively adopted by the industry. GEDCOM is an open format with a simple lineage-linked structure, in which each record relates to either an individual or a family, and relevant information, such as names, events, places, relationships, and dates, appears in a hierarchical structure [36]. There are several open online GEDCOM databases, including GenealogyForum [35], WikiTree[8], GedcomIndex[9], Anu Museum[10], Ancestry.com, and others.

In GEDCOM format, every person (individual) in the family tree is represented as a node that may contain known attributes, such as first name, last name, birth date and place, death date and place, burial date and place, notes, occupation, and other information. Two individuals are not linked to one another directly. Each individual is linked to a family node as a "spouse" (i.e., a parent) or a "child" in the family. Figure 2 shows a sub-graph corresponding to a Source Person (SP) whose data is presented in the GEDCOM file in Figure 3. Each individual and family are assigned a unique ID – a number bracketed by @ symbols and a class name (INDI – individual, FAM – family). The source person is noted as SP (@I137@ INDI - Emily Williams in the GEDCOM file), families as F and other persons as P. In this example, P3, P4, P5, and P6 are the grandparents of SP; P1 and P2 are SP's parents in family F1 (@F79@ in the GEDCOM file); P7 and P8 are SP's siblings; P10 (@I162@ INDI – John Williams in the GEDCOM file) is SP's spouse from family F4 (@F73@ in the GEDCOM file), P12 and P13 are SP's children; and P15, P16, and P17 are SP's grandchildren. Moreover, as seen in Figure 3, SP was a female, born on 28 MAY 1816 in New York, USA, who died on 7 FEB 1899 in Uinta, Wyoming, USA, and was buried three days later in the same place. Furthermore, SP was baptized on 1 JUN 1832 and was endowed on 30 DEC 1845 in TEMP NAUVO (maybe Nauvoo Temple[11], Illinois). Her husband, P10, John Williams, was a male, born on 16 MAY 1826 in Indiana, USA, who died on 25 SEP 1912 in Uinta, Wyoming, USA, and was buried three days later in the same place. He was baptized on 9 AUG 1877, although there is a note stating that it may be on the 12 of AUG 1877, and he was endowed with his wife. For practical reasons, the GEDCOM file example in Figure 3 contains only a small portion of the data presented in Figure 2.

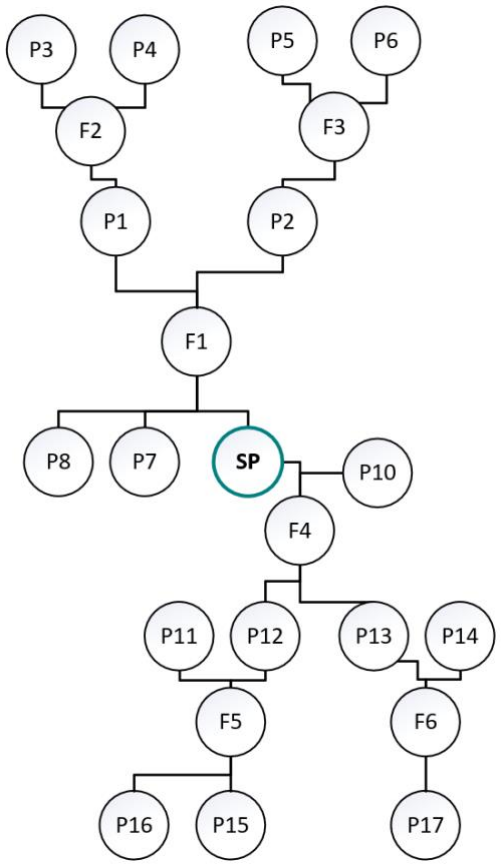

Fig. 2. Family tree structure.

[8] https://www.wikitree.com/

[9] http://gedcomindex.com/gedcoms.html

[10] https://dbs.anumuseum.org.il

[11] https://churchofjesuschristtemples.org/nauvoo-temple/

```
0 HEAD
1 SOUR SomeSite
2 NAME Some Site
2 VERS 3.0
1 DATE 30 JUN 1985
2 TIME 19:38:50
1 FILE example.ged
1 GEDC
2 VERS 5.5
2 FORM LINEAGE-LINKED
...
1 CHAR ANSEL
0 @I137@ INDI
1 NAME Emily Williams
1 SEX F
1 BIRT
2 DATE 28 MAY 1816
2 PLAC New York, USA
1 DEAT
2 DATE 7 FEB 1899
2 PLAC Uinta, Wyoming, USA
1 BURI
2 DATE 10 FEB 1899
2 PLAC Uinta, Wyoming, USA
1 BAPL
2 DATE 1 JUN 1832
1 ENDL
2 DATE 30 DEC 1845
2 TEMP NAUVO
1 FAMS @F73@
1 FAMC @F79@
1 SLGC
2 DATE 18 NOV 1894
2 TEMP SLAKE
1 CHAN
2 DATE 14 MAY 1999
3 TIME 09:57:42
0 @I162@ INDI
1 NAME John Williams
1 SEX M
1 BIRT
2 DATE 16 MAY 1826
2 PLAC Indiana, USA
1 DEAT
2 DATE 25 SEP 1912
2 PLAC Uinta, Wyoming, USA
1 BURI
2 DATE 28 SEP 1912
2 PLAC Uinta, Wyoming
1 BAPL
2 DATE 9 AUG 1877
1 ENDL
2 DATE 30 DEC 1845
2 TEMP NAUVO
1 FAMS @F73@
1 FAMC @F1598@
1 NOTE Baptism date appears to be 3 days later by the records of the city…
...
```

Fig. 3. (part of the) GEDCOM family tree file.

2.2. Question answering using DNN

A DNN is a computational mathematical model that consists of several "neurons" arranged in layers. Each neuron performs a computational operation and transmits the computed information (calculation result) to the neurons in the next layer. The information is passed over and changed from layer to layer until it becomes the output in the network's last layer. The conventional learning method is backpropagation, which refers to learning as an optimization problem [122]. After each training cycle, a comparison between the network prediction (output) and the actual expected result is performed, and a "loss" (i.e., the gap) is calculated to estimate the changes needed in the network operations (the weight of neuron's transformation). Changes in the network weights are usually performed using the Gradient Descent methods [7].

In recent years, DNNs have become the state-of-the-art method for text analysis in the cultural heritage space [110], and natural language question-answering systems based on DNN have become the state-of-the-art method for solving the question answering task [61]. The underlying task of question answering is Machine Reading Comprehension (MRC), which allows machines to read and comprehend a specified context passage for answering a question, similarly to language proficiency exams. Question answering, on the other hand, aims to answer a question without a specific context. These QA systems store a database containing a sizeable unstructured corpus and generate the context in real-time based on relevant text passages to the input question [138]. Due to the magnitude of comparisons needed between the query and each text passage in the corpus, and due to the number of calculations (a large number of multiplications of vectors and matrices) when a DNN model predicts the answer span for every given text passage, DNNs are not applied on the entire database of texts, but only on a limited number of passages. Hence, when a user asks a question, the system searches [12] the database for K passages that are relevant to the user question. The system will then use the DNN model to predict the answer span (start and end positions) for each text passage (from the K passages) with a confidence level. The answer with the highest confidence level is selected as the answer to be presented to the user. Thus, a typical pipeline (shown in Figure 4) of DNN for question answering will be a compound of (1) two inputs - (a) a text passage (i.e., a document) that may contain the answer, and (b) a question; and (2) two outputs: (a) the start index of the answer in the text passage, and (b) the end index of the answer in the text passage. The inputs are encoded into vectors using static embeddings methods, such as

[12] A common approach for finding relevant passages is reverse indexing [11, 53, 54, 71, 104]

Word2Vec [77] and GloVe [86] or using contextualized embeddings of words like Bidirectional Encoder Representations from Transformers (BERT) [22], Embeddings from Language Models (ELMo) [87] and other methods [88]. One of the main advantages of contextual embeddings is the ability to handle disambiguations of words and entities [81, 129]. The input vector is transferred through the network, and the final layer output vectors are the probability of every word to be the start or the end of the span (i.e., answer). The score of every span is a combination of the start and end tokens' probabilities. The most probable span is then translated back to a sequence of words using the embedding method [22] (see section 3.2 for a more detailed description). Researchers proposed various DNN-based models to solve the task of finding (ranking) an answer span (the part of the text that contains the answer for the question) in the document [22, 97, 118, 119, 133] or a single sentence [34, 62].

2.2.1. Natural question answering using DNN architecture

Over the years, different deep learning layers have been developed with various abilities. Until recently, the typical architecture for natural language questions answering was based on Recurrent Neural Networks (RNN) such as Long Short Term Memory (LSTM) [48] and Gated Recurrent Units (GRU) layers [17]. RNN layers allow the network to "remember" previously calculated data and thus learn answers regarding an entire sequence. These layers are used to construct different models, including a sequence-to-sequence model [112] that uses an encoder-decoder architecture [17] that fits the question-answering task. This model maps a sequence input to a sequence output, like a document (sequence of words) and a question (sequence of words) to an answer (sequence of words) or to classify words (whatever the word is the start or the end of the answer). RNN architecture often works with direct and reverse order sequences (bidirectional-RNN) [96]. It may also include an attention mechanism [115], which "decides" (i.e., ranks) which parts in the sequence are more important than others during the transformation of a sequence from one layer to another.

Another typical architecture is based on a Convolutional Neural Network (CNN). Unlike RNNs, CNNs architecture does not have any memory state that accumulates the information from the sequence data. CNN architecture uses pre-trained static embeddings where each CNN channel aggregates information from the vectorial representation. Channels of different sizes enable it to deal with n-gram-like information in a sentence [57].

Question answering task can also be modeled as a graph task (e.g., traversal, subgraph extraction). The data can be represented as a knowledge graph (KGQA), where each node is an entity, and each edge is a relation between two entities. When answering the question, the algorithm finds the relevant entities for the question and traverses over the relations or uses the node's attributes to find the answer node or attribute [13, 24, 134]. To work with graphs, Graph Neural Networks (GNN) [94] models have been developed that operate directly on the graph structure. GNN can be used for resolving answers directly from a knowledge graph by predicting an answer node from question nodes (i.e., entities) [29, 38, 72, 80, 101, 134]. The GNN model is similar to RNN in the sense that it uses near nodes and relations (instead of previous and next token in RNN) to classify (i.e., label) each node. However, these models cannot directly work with unstructured or semi-structured data or rely on the ability to complete and update the knowledge graph from free texts using knowledge graph completion tasks, such as relation extraction [8, 82, 128] or link prediction [32, 52].

An improved approach considered to be the state-of-the-art in many NLP tasks, including question answering, is Transformers architecture [115], which uses the attention mechanism with feed-forward layers (not RNNs); this kind of attention is also called Self Attention Network (SAN). Well-known examples of SANs are Bidirectional Encoder Representations from Transformers (BERT) [22] and GPT-2 [89] models. Several BERT-based models were developed in recent years [125], achieving state-of-the-art performance (accuracy) in different question answering tasks. These include RoBERTa - a BERT model with hyperparameters and training data size tuning [70]; DistilBERT - a smaller, faster, and lighter version of BERT [93]; ELECTRA – a BERT-like model with a different training approach [18]. Although standard BERT-based models receive textual sequence as input, all the above architectures can also be mixed. For example, a Graph Convolutional Network (GCN) [114] can be utilized for text classification by modeling the text as a graph and using the filtering capabilities of a CNN [131].

There are several question-answering DNN pipelines based on knowledge graphs that support semi-structured data (a mix of a structured graph and unstructured texts) [29, 40, 134, 137]. As shown in Figure 5, a current state-of-the-art pipeline of this type,

Deciphering Entity Links from Free Text (DELFT) [134], uses the knowledge graph to extract related entities and sentences, filters possible textual sentences using BERT, and then traverses a filtered subgraph using a GNN. The pipeline starts with identifying the entities in the question. Then, related entities ("candidates") from the knowledge graph and relevant sentences ("evidence relations") from unstructured texts are extracted and filtered using BERT. A new subgraph is generated using the question entities, the filtered evidence relations, and the candidate entities. Using this subgraph, a GNN model learns to rank the most relevant node. Thus, the model obtains a "trail" from the question nodes to a possible candidate node (i.e., answer). The pipeline applies two DNN models: a BERT model to rank the evidence relations and a GNN model to traverse the graph (i.e., predict the answer node).

However, these methods, using the unstructured texts to create or complete the knowledge graph, rely heavily on well-defined semantics and fail to handle questions with entities completely outside the knowledge graph or questions that cannot be modeled within the knowledge graph. For example, Differentiable Neural Computer (DNC) [38] can be used to answer traversal questions ("Who is John's great-great-grandfather?"), but not to answer content-related questions when the answer is written in the person's bio notes (e.g., "When did John's great-great-grandfather move to Florida?"). As part of the evaluation experiments in this study, the performance of the above mentioned DELFT pipeline, adapted to the genealogical domain, was compared to that of the proposed pipeline.

In summary, the generic question answering pipelines described above cannot be applied as-is in the genealogical domain, without compromising on accuracy, for the following reasons: (1) The raw data is structured as graphs, each graph contains more information than a DNN model can handle in a single inference process (each node is equivalent to a document), (2) A user may ask about different nodes and different scopes of relations (i.e., different genealogical relation degrees); (3) There is a high number of nodes containing a relatively small volume of structured data and a relatively large volume of unstructured textual data. In addition, the vast amount of different training approaches, hyperparameters tuning, and architectures indicate the complexity of the models and sensitivity to a specific domain and sub-task.

The question answering approach proposed in this study simplifies the task pipeline by converting the genealogical knowledge graph into text, which is then combined with unstructured genealogical texts and processed by BERT's contextual embeddings. Converting the genealogical graph into text passages can be performed using knowledge-graph-to-text templates and methodologies [21, 26, 55, 76, 123], and knowledge-graph-to-text machine learning and DNN models [5, 33, 63, 66, 68, 78, 79, 99, 106]. Template-based knowledge-graph-to-text methods use hardcoded or extracted linguistic rules or templates to convert a subgraph into a sentence. Machine learning and DNN models can be trained to produce a text from knowledge-graph nodes. The input for a knowledge-graph-to-text model is a list of triples of two nodes and their relation, and the output is a text passage containing a natural language text with input nodes and their relations as syntactic sentences. To this end, DNN models are often trained using commonsense knowledge graphs of facts, such as ConceptNet [107], BabelNet [83], DBpedia [3], and Freebase [85], where nodes are entities, and the edges represent the semantic relationships between them. Some models use the fact that knowledge graphs are language-agnostic to generate texts in multi-languages (e.g., [79]).

2.3. Questions and answers generation for DNN-based question answering systems

Training of a DNN question answering model requires a set of text passages and corresponding pairs of questions and answers. Multiple approaches exist for generation of questions (and answers): knowledge-graph-to-question template-based methodology (similar to the context generation) [67, 98, 136, 140], WH questions (e.g., Where, Who, What, When, Why) rule-based approach [80], knowledge graph-based question generation [16, 50], and DNN-based models for generating additional types of questions [25, 49, 117, 135]. The rule-based method uses part-of-speech parsing of sentences using the Stanford Parser [59], creates a tree query language and tree manipulation [65], and applies a set of rules to simplify and transform the sentences to a question. To guarantee question quality, questions are ranked by a logistic regression model for question acceptability [44]. The DNN question generation models are trained on SQuAD [90, 91] or on facts from a knowledge graph to predict the question and its correct answer from the context (i.e., the opposite task from question answering) using bi-directional [96] LSTM [48] encoder-decoder [17] model with attention [115].

This study adopted the format of the SQuAD dataset, which is a well-known benchmark for machine learning models on question answering tasks with a formal leaderboard[13]. SQuAD is a reading comprehension dataset consisting of questions created by crowd workers on a set of Wikipedia articles. The answers to the questions are segments of text from the corresponding reading passage (context), or the question might be unanswerable. SQuAD 2.0 combines 100,000 questions and answers and over 50,000 unanswerable questions written adversarially by crowd workers to look similar to answerable ones. To do well on SQuAD 2.0, natural question answering models must answer questions when possible and determine when no answer is supported by the paragraph, in which case they must abstain from answering.

SQuAD 2.0 is a JSON formatted dataset, presented in Figure 6, where each topic (a Wikipedia article) has a *title* and *paragraphs*. Each paragraph contains a *context* (text passage) and questions (*qas*). Each question contains the *question* text, *id*, may contain *answers* (if it is answerable), may contain *plausible answers*, or be marked as *impossible*. Each answer is constructed from a *text* and a *start index* (the word index) of the answer in the text passage.

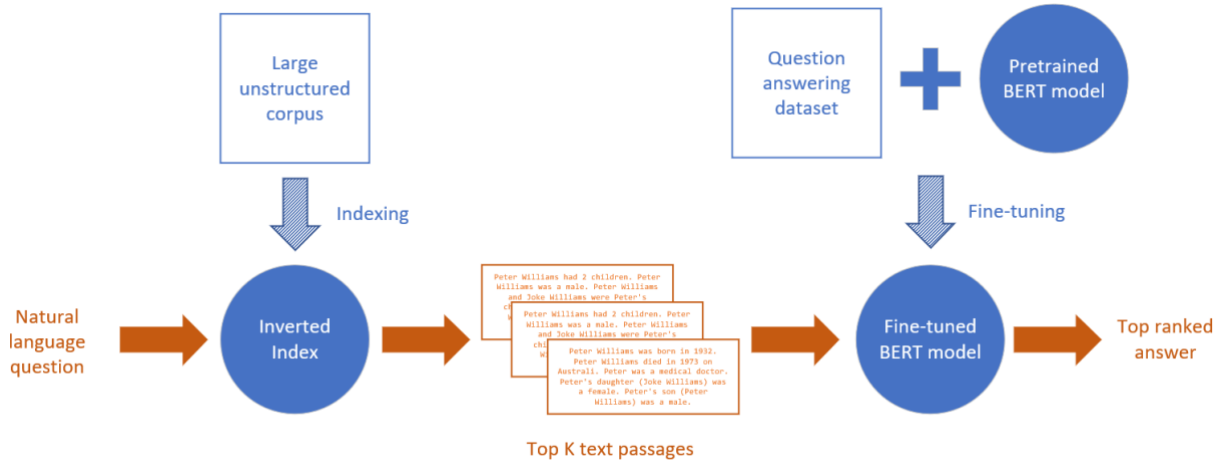

Fig. 4. Typical open-domain question answering pipeline.

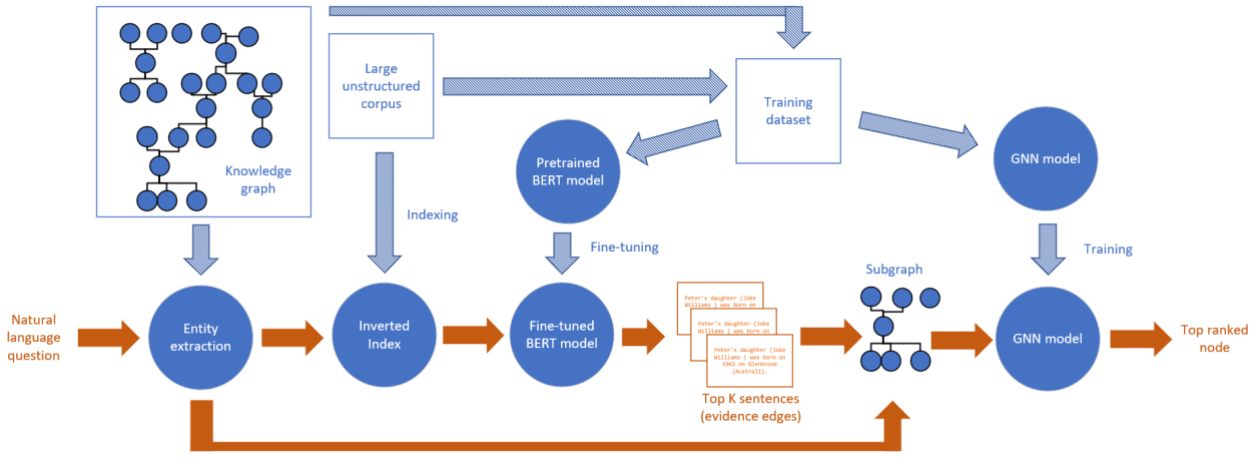

Fig. 5. Typical knowledge graph question answering pipeline.

[13] https://rajpurkar.github.io/SQuAD-explorer/

```
{ "version": "v2.0",
  "data": [
  { "title": "Normans",
    "paragraphs": [
        { "qas": [
            { "question": "Who was the duke in the battle of Hastings?",
        "id": "56dddf4066d3e219004dad5f",
        "answers": [{ "text": "William the Conqueror", "answer_start": 1022}],
        "is_impossible": false },
        { "question": "Who ruled the duchy of Normandy",
        "id": "56dddf4066d3e219004dad60",
        "answers": [ { "text": "Richard I", "answer_start": 573 }],
        "is_impossible": false },
        { "plausible_answers": [
         { "text": "political, cultural and military",
           "answer_start": 31 }],
        { "question": "What type of major impact did the Norman dynasty have on modern Europe?",
        "id": "5ad3a266604f3c001a3fea27",
        "answers": [],
        "is_impossible": true},
       … ],
      "context": "The Norman dynasty had a major political, cultural and military impact on medieval Europe and even
the Near East. The Normans were famed for their martial spirit and eventually for their Christian piety, becoming
exponents of the Catholic orthodoxy into which they assimilated. They adopted the Gallo-Romance language of the
Frankish land they settled, their dialect becoming known as Norman, Normaund or Norman French, an important literary
language. The Duchy of Normandy, which they formed by treaty with the French crown, was a great fief of medieval
France, and under Richard I of Normandy was forged into a cohesive and formidable principality in feudal tenure. The
Normans are noted both for their culture, such as their unique Romanesque architecture and musical traditions, and
for their significant military accomplishments and innovations. Norman adventurers founded the Kingdom of Sicily
under Roger II after conquering southern Italy on the Saracens and Byzantines, and an expedition on behalf of their
duke, William the Conqueror, led to the Norman conquest of England at the Battle of Hastings in 1066..."
     },
     … ]
   },
    … ]
 }
}
```


Fig. 6. SQuAD 2.0 JSON format example.

3. Methodology

While using DNNs for the open-domain question answering task has become the state-of-the-art approach, automated question answering systems for genealogical data is still an underexplored field of research. This paper presents a new methodology for a DNN-based question answering pipeline for semi-structured heterogeneous genealogical knowledge graphs. First, a training corpus that captures both the structured and unstructured information in genealogical graphs is generated. Then, the generated corpus is used to train a DNN-based question answering model.

3.1. Gen-SQuAD generation and graph traversal

The first phase in the proposed methodology is to generate a training dataset using the text sequence encoding with a graph traversal algorithm. This dataset should contain questions with answers and free text passages from which the model can retrieve these answers.

Generating a training dataset from genealogical data is a three-step process. The result of the process is Gen-SQuAD, a SQuAD 2.0 format dataset tailored to the genealogical domain. As shown in Figure 7, the process includes the following steps: (1) decomposing the GEDCOM graphs to CIDOC-CRM-based [14] knowledge sub-graphs, (2) generating text passages from the obtained knowledge sub-graphs, and (3) generating questions and answers from the text passages. Finally, the context and matching questions and answers are saved in the SQuAD 2.0 JSON format. The following sections present in detail each step of the Gen-SQuAD generation process.

[14] http://www.cidoc-crm.org/

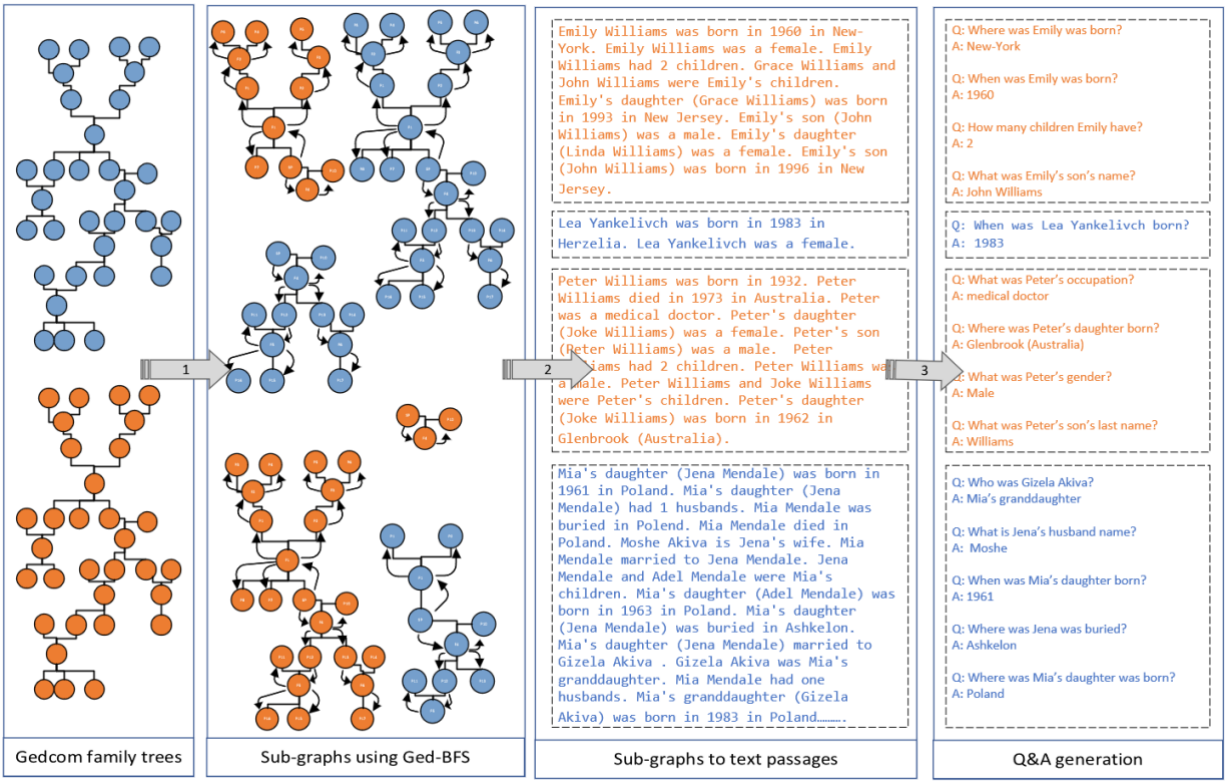

Fig. 7. Gen-SQuAD generation.

3.1.1. Sub-graph extraction and semantic representation

While there are some DNN models that can accept large inputs [9, 58], due to computational resource limitations, many DNN models tend to accept limited size inputs, usually ranging from 128 to 512 tokens (i.e., words) [141]. However, family trees tend to hold a lot of information, from names, places, and dates to free-text notes, life stories, and even manifests. Therefore, using the proposed methodology, it is not practical to build a model that will read an entire family tree as an input (sequence), and it is necessary to split the family tree into sub-trees (sub-graphs). Several generic graph traversal algorithms may be suitable for traversing a graph and extracting sub-graphs, such as Breadth-First-Search (BFS) and Depth-First-Search (DFS). BFS's scoping resembles a genealogy exploration process that treats first relations between individuals that are at the same depth level (relation degree) in the family tree, moving from the selected node's level to the outer level nodes. However, the definition of relation degrees in genealogy (i.e., consanguinity) is different from the pure graph-theory mathematical definition implemented in BFS [12]. For example, parents are considered first-degree relations in genealogy (based on the ontology), while they are considered to be second-degree relations mathematically, since there is a family node between the parent and the child (i.e., the parent and the child are not connected directly), with siblings considered to be second-degree relations in both genealogy and graph theory. Combined BFS-DFS algorithms such as Random Walks [39] do not take into account domain knowledge and sample nodes randomly. In the genealogical research field, several traversal algorithms have been suggested for user interface optimization [56]. However, these algorithms aim to improve interfaces and user experience and are not suitable for complete data extraction (graph to text) tasks.

This paper presents a new traversal algorithm, Gen-BFS, which is essentially the BFS algorithm adapted to the genealogical domain. The Gen-BFS algorithm is formally defined as follows:

Algorithm 1

Gen-BFS algorithm.

Input: Node (*SP*), Depth (*D*)
Output: Traverse queue (*TQ*)
Initialization: Node queue (*NQ*), Depth queue (*DQ*), Current depth (*CD* = 0), Nodes to depth increase (*NTDI* = 1), Next nodes to depth increase (*NNTDI* = 0)

```
1.  NQ enqueue SP
2.  while NQ is not empty
3.      n = NQ dequeue
4.      DQ enqueue n
5.      if n is Person
6.          kn = n→{fam_child} union n→{fam_parent}
7.      else
8.          kn = n→{child_fam}  union n→{parent_fam}
9.      NNTDI = NNTDI + count (kn not in NQ)
10.     NTDI = NTDI – 1
11.     if NTDI = 0
12.         if n is Person
13.             CD = CD + 1
14.             if CD > D
15.                 break while
16.         NTDI = NNTDI
17.         NNTDI = 0
18.     for n in kn
19.         if n not in NQ
20.             NQ enqueue n
21. while DQ is not empty
22.     dn = DQ dequeue
23.     TQ enqueue dn
24.     if dn is Person
25.         for f in dn→{fam_parent}
26.             for p in f→{parent_fam}
27.                 if p not in DQ and p not in TQ
28.                     TQ enqueue p
29. return TQ
```

Where each node can be a Person or a Family, each Person node has two links (edges) types: fam_{child} (FAMC in GEDCOM standard) and fam_{parent} (FAMS in GEDCOM standard), each Family has the opposite edge types: $child_{fam}$ and $parent_{fam}$. Where $\{fam_{child}\}$ is the collection of all the families in which a person is considered a child (biological family and adopted families), $\{fam_{parent}\}$ is the collection of all the families in which a person is a parent (spouse) (i.e., all the person's marriages), $\{child_{fam}\}$ is a collection of all the persons that are considered to be children in a family and $\{parent_{fam}\}$ is a collection of all the persons considered to be a parent in a family. For example, the SP in Figure 2 is linked to two nodes. The link type to F1 is fam_{child}, and the link type to F4 is fam_{parent}. The family F1 in Figure 2 has two types of links. The link type to SP, P7, P8 is $child_{fam}$, and the link type to P1 and P2 is $parent_{fam}$.

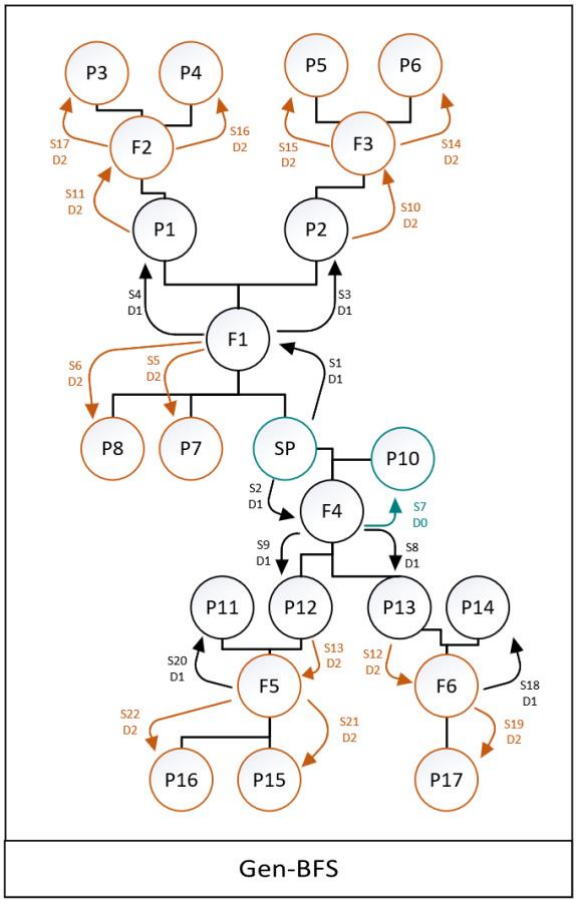

Fig. 8. Gen-BFS algorithm[15].

Figure 8 illustrates the Gen-BFS traversal applied to the family tree presented in Figure 2. As shown in Figure 8, Gen-BFS is aware of the genealogical meaning of the nodes and reduces the tree traversal's logical depth. It ignores families in terms of relation degree, considers SP's spouses as the same degree as SP and SP's parents and children as first degree, and keeps siblings and grandparents as second-degree. In particular, lines 1-20 in Algorithm 1 represent a BFS-style traverse over the graph. In lines 5-8, the algorithm introduces domain knowledge and adds nodes to its queue according to the node type. The code in lines 9-17 ensures that the traversal will stop at the desired depth level. If the current node is a Person (line 12) and the current depth (CD) is about to

[15] An algorithm step is noted as S. The degree of relation is noted as D. Relations are color-coded as follows: Zero-degree relation (self) - turquoise, First-degree relations – black, and Second-degree relations – brown.

get deeper than the required depth (D), then the while loop will end (line 14). Otherwise, the Persons and Families in the current depth (kn) will be added to the node queue (NQ) and may (depending on the stop mechanism) be added to the depth queue (DQ). In line 21, the depth queue (DQ) holds all the Family nodes and most of the Person nodes (except for spouses of the last depth level's Person nodes) within the desired depth level. For example, traversing with D = 1 over the family tree in Figure 2 will result in DQ that contains SP and her children and parents (F1, F4, P10, P1, P2, P12, and P13). However, according to the genealogical definition of depth levels in a family relationship, the children's spouses, P11 and P14 (but not the grandchildren, F5 and F6, which belong to D = 2) should also be retrieved. Lines 21-28 address this issue and add the missing Person nodes, thus logically reducing the depth of the graph.

Each family tree was split into sub-graphs using the Gen-BFS algorithm. New sub-graphs were created for each person as SP (source person) and its relations at different depth levels. Therefore, there is an overlap between the sub-graphs (a person can appear in several sub-graphs), and the sub-graphs cover all the individuals and relations in a given family tree. The Gen-BFS traversal algorithm is used both for dataset generation and for selecting the scope of the user's query in the inference phase (i.e., when answering the question).

Once extracted, each genealogical sub-graph was presented as a knowledge graph. This study adopted an event-based approach to data modeling presented in the past literature ([2, 31, 113]). As in [113], a formal representation of the GEDCOM heterogeneous graph (excluding the unstructured texts) as a knowledge graph was implemented using CIDOC-CRM, but in a more specific manner (e.g., we used concrete events and properties such as *birth, brought into life* as opposed to [113] that used generic vocabulary). We chose to use CIDOC-CRM as it is a living standard (ISO 21127:2014) for cultural heritage knowledge representation. CIDOC-CRM is designed as "a common language for domain experts" and "allows for the integration of data from multiple sources in a software and schema-agnostic fashion" [60]. It has been applied as a base model and extended in many domains related to cultural heritage, and in this study, it was chosen as a basis for defining the genealogical domain ontology due to its standard and generic nature and event-based structure, that enables *n*-ary rather than binary relationships between entities in the ontology, as required for representing genealogical and biographic data based on events in families and person's lives (e.g., E67 represents a birth event that connects a person, a place and a time span). Genealogical graphs contain instances of two explicit classes: Person (E21 in CIDOC-CRM) and family that can be represented as a Group (E74 in CIDOC-CRM); and several implicit classes: Place (E53), Event (E5), Death (E69), Birth (E67) and others. These implicit classes are not structured as separate entities in the GEDCOM standard, but need to be extracted from the GEDCOM attributes. Properties matching various GEDCOM relations can also be easily found in CIDOC-CRM, e.g., the relation of a person to its children can be represented using P152 (is parent of).

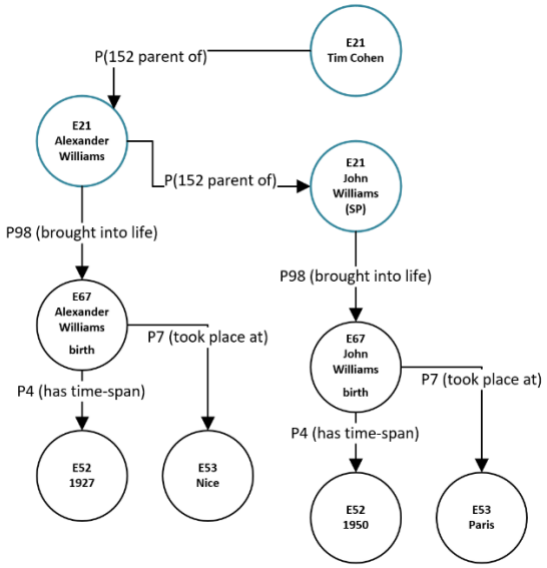

Fig. 9. GEDCOM individual's knowledge graph in the CIDOC-CRM-based format.

Figure 9 is an example of a representation of the GEDCOM sub-graph as a knowledge graph. As illustrated in the figure, the SP node is an instance of the class Person and has a relation (property) to a birth event (E21=>P98=>E67) with a relation to the place, Paris (E67=>P7=>E53) and a relation to the birth year with the value 1950 (E67=>P4=>E52). Representing GEDCOM as a knowledge graph is a critical step as the dataset generation method is based on well-established knowledge-graph algorithms, as described next.

3.1.2. Text passage generation

Next, a textual passage from each sub-graph is generated, representing the *SP's* genealogical data based on the graph-to-sequence. Text passages were

generated using a knowledge-graph-to-text DNN model [68] and completed (for low model confidence or missing facts) with knowledge-graph-to-text template-based methodology [76]. It is important to note that converting the obtained genealogical knowledge sub-graphs to text is a more straightforward task than the open domain knowledge-graph-to-text or generic commonsense knowledge-graph-to-text task, since they are well structured and relatively limited in their semantics. For example, the sub-graph presented in Figure 9 can be converted to a sentence with template rules or using DNN models. A rule example will be: [First Name] [Last Name] *was born in* [Birth Year] *in* [Birthplace] = "John Williams was born in 1950 in Paris".

Using a knowledge-graph-to-text DNN model [68] and a knowledge-graph-to-text templates methodology [76], multiple variations of sentences conveying the same facts (comprised of the same nodes and edges in the graph) were composed based on different templates and combined with the sentence paraphrasing using a DNN-based model (the model of [63]). Most of the text passages were generated using a DNN model. However, the template-based method added variations that the DNN model did not capture. Table 1 above presents examples of such sentences created for the sub-graph in Figure 9.

Another critical challenge resolved by this approach is the multi-hop question answering problem, where the model needs to combine information from several sentences to answer the question. Although there are multi-hop question answering models presented in the literature [30, 74], their accuracy is significantly lower than a single-hop question answering. To illustrate the problem, consider a user asking about the SP's (John's) grandfather: "Where was John's grandfather born?" or "Where was Tim Cohen born?", where Tim Cohen refers to John's grandfather. To answer both questions without multi-hop reasoning for resolution of multiple references to the same person, the graph-to-text template-based rules include patterns that encapsulate both the SP's relationship type (John's grandfather) and the relative's name (Tim Cohen), thus allowing the model to learn that Tim Cohen is John's grandfather. There are three types of references to a person that allows the DNN model to resolve single or multi-hop questions: 1) Direct referencing to a person with his/hers first and last name (e.g., John Williams), 2) Partial referencing to a person with his/hers first or last name (e.g., John), and 3) Multi-hop encapsulation, i.e., referencing to a person with their relative name to the SP (e.g., Alexander's son).

As a result of the above processing, multiple text passages were created for each SP's sub-graph. Since each sentence is standalone and contains one fact, sentences were randomly ordered within each text passage. Thus, even if the passage is longer than the neural model's computing capability, the model will likely encounter all types of sentences during its training process. These text passages were further encoded as vectors (i.e., embeddings) to train a DNN model that learns contextual embeddings to predict the answer (i.e., start and end positions in the text passage) for a given question.

Table 1

Genealogical-knowledge-graph-to-text context template example.

Template-based rule example	**Result**	**Reference type**
[First Name] [Last Name] was born in [Birth Year] in [Birthplace]	John Williams was born in 1950 in Paris	Direct
[First Name] was born in [Birth Year] in [Birthplace]	John was born in 1950 in Paris	Partial
[Name relative of SP] ([First Name] [Last Name]) was born in [Birth Year] in [Birthplace]	Alexander's son (John Williams) was born in 1950 in Paris	Multi-hop encapsulation
[First Name] was born in [Birthplace] in [Birth Year]	John was born in Paris in 1950	Partial
[Relative First Name] [Relative Last Name] ([Relation to SP]) was born in [Birth Year] in [Birthplace]	Alexander Williams (John's father) was born in 1927 in Nice.	Multi-hop encapsulation
In [Birth Year] [First Name] was born	In 1950 John was born	Partial
[Birthplace] was [First Name] 's birthplace	Paris was John's birthplace	Partial

3.1.3. Generation of questions and answers

Using the generated text passages (contexts), pairs of questions and answers were created. The answers were generated first, and then the corresponding questions were built for them as follows. Knowledge graph nodes and properties (relationships), as well as named entities and other characteristic keywords extracted from free text passages were used as answers. To achieve extensive coverage, multiple approaches were used for generation of questions. First, a rule-based approach was applied for question generation from knowledge graphs [140] and a statistical question generation technique [44] was utilized for WH question generation from the unstructured texts in GEDCOM.

Most of the questions (73%) were created using these methods. To identify the types of questions typical of the genealogical domain and define rule-based templates for their automatic generation, this study examined the genealogical analysis tasks that users tend to perform on genealogical graphs [10]. These tasks include: (1) identifying the SP's ancestors (e.g., parents, grandparents) or descendants (e.g., children, grandchildren), (2) identifying the SP's extended family (second-degree relations), (3) identifying family events, such as marriages, (4) identifying influential individuals (e.g., by occupation, military rank, academic achievements, number of children), and (5) finding information about dates and places, such as the date of birth, and place of marriage [4, 10]. These analysis tasks were adopted to define characteristic templates for natural language questions that a user may ask about the SP or its relatives. Some of these questions can be answered directly from the structured knowledge graph (e.g., “When was Tim’s father born?”), while others can only be answered using the unstructured texts attached to the nodes (e.g., “Did Tim’s father have cancer?”).

A DNN-based model for generating additional types of questions [25] was used to complement the rule-based method. The neural question generation model predicted questions from all the unstructured texts in the GEDCOM data and produced 24% of the questions in the dataset (excluding duplicate questions already created using the WH-based and rule-based approaches).

Table 2

Knowledge-graph-to-text question template examples.

Template-based rule example	**Result**
How many children did [First Name] [Last Name] have?	How many children did John Williams have?
How many grandchildren did [Relative First Name] [Relative Last Name] ([Relation to SP]) have?	How many grandchildren did Alexander Williams (John's father) have?
Was [Birthplace] [First Name] 's birthplace?	Was Paris John's birthplace?

Finally, additional rules were manually compiled using templates [1, 28] to create questions missed by previous methods, mainly quantitative and yes-no questions (as illustrated in Table 2). These questions were 3% of all the questions in the datasets. All answer indexes were tested automatically to ensure that the answer text exists in the context passage. A random sample of 120 questions was tested manually by the researchers as a quality control process, and the observed accuracy was virtually 100%. However, it is still possible that DNN generated some errors. Nevertheless, even in this case, the study's conclusions would not change, as such errors would have a similar effect (same embeddings) on all the tested models.

3.2 Fine-tuning the BERT-based DNN model for question answering

Fine-tuning a DNN model is the process of adapting a model that was trained on generic data to a specific task and domain [22]. An initial DNN model is usually designed and trained to perform generic tasks on large domain-agnostic texts, like Wikipedia. In the case of the open-domain question answering, the BERT baseline model was pre-trained on English Wikipedia and Books Corpus [139] using Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives [22]. The MLM methodology is a self-supervised dataset generation method. For each input sentence, one or more tokens (words) are masked, and the model's task is to generate the most likely substitute for each masked token. In this fill-in-the-blank task, the model uses the context words surrounding a mask token to try to predict what the masked word should be. The NSP methodology is also a self-supervised dataset generation method. The model gets a pair of sentences and predicts if the second sentence follows the first one in the dataset. MLM and NSP are effective ways to train language models without annotations as a basis for various supervised NLP tasks. Combining MLM and NSP training methods allow modeling languages with both word-level relations and sentence-level relations understanding. The pre-trained BERT-based question answering model was designed with 12 layers, 768 hidden nodes, 12 attention heads, and 110 million parameters. Using such a pre-trained model, DNN layers can be added to fit to a specific task [22].

As shown in Figure 10, a new BERT-based model, Uncle-BERT, was fine-tuned for genealogical question answering as follows: (1) adding a pair of output dense layers (vectors) with dimensions of the hidden states in the model (*S* and *E)*, (2) computing the probability that each token in these layers (vectors) is the start (*S*) or end (*E*) of the answer, and finally (3) running and tuning the baseline BERT model described above for learning *S* and *E*. The probability of a token being the start or the end of the answer is the dot product between the token's numerical representation (i.e., embeddings) in the last layer of BERT and the new output layers (vectors *S* or *E*), followed by a softmax activation over all the tokens. Then, using the genealogical training dataset, the model is trained to solve the task in the study's domain. It should be noted that generation methods for pre-trained static node embeddings like node2vec [39] or TransE [14] treat triples as the training instance for embeddings, which may be insufficient to model complex information transmission between nodes. Therefore, the information is encoded from graph nodes into syntactic sentences and then the original BERT approach [22] is applied to generate comprehensive contextual embeddings from these sentences [43].

Figure 11 summarizes the developed genealogical question answering pipeline. To simplify the task, the proposed architecture asks the user to first select the family tree from the corpus (future research can eliminate this step by embedding the family trees [37] and ranking them based on similarity to the question [92]). As demonstrated in the figure, the family tree corpus (comprised of GEDCOM files) is processed into question answering datasets for different scopes. The process starts when a user selects a specific person from a family tree. Then the user indicates a scope (a genealogical relation degree, as described in Figure 1) to ask about (e.g., the *SP* itself, first-degree relative, second-degree relatives) and asks a question ("What was Alexander's father's military rank?"). The Gen-BFS algorithm incorporates the SP and the scope to generate a text passage that encapsulates the *SP*'s scope aligned with the user intent (equivalent to finding the top K text passages in the open-domain question answering pipeline). Finally, a fine-tuned DNN model, selected based on the requested relational degree (i.e., a model trained to predict answers on the requested relational degree), predicts the answer using the generated text passage and a question as inputs.

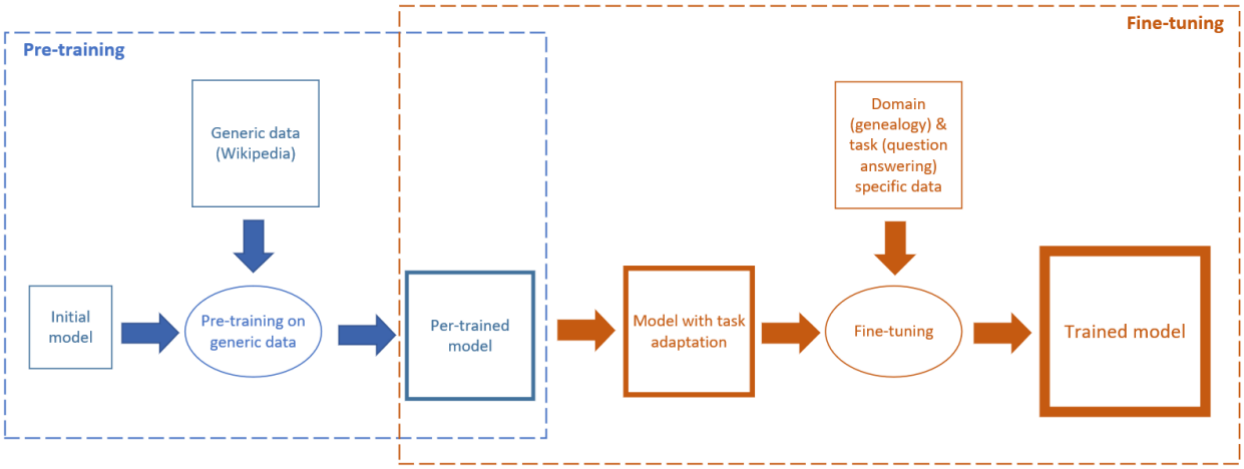

Fig. 10. The DNN model fine-tuning process.

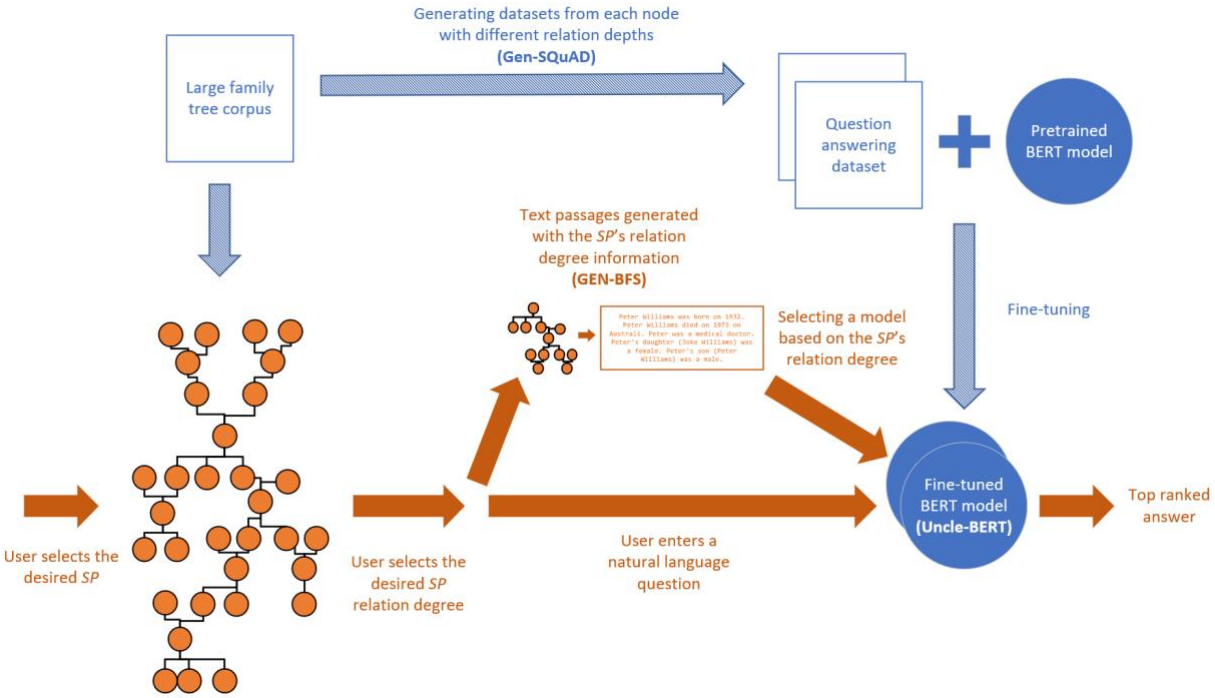

Fig. 11. Genealogical question answering pipeline (the proposed architecture).

4. Experimental design

This section describes the experimental dataset and training conducted to validate the proposed methodology for the genealogical domain.

4.1. Datasets

In this research, 3,140 family trees containing 1,847,224 different individuals from the corpus of the Douglas E. Goldman Jewish Genealogy Center in Anu Museum [16] were used. The Douglas E. Goldman Jewish Genealogy Center contains over 5 million individuals and over 30 million family tree connections (edges) to families, places, and multimedia items. To comply with the Israeli privacy regulation[17] and the European general data protection regulation[18] (GDPR), only family trees for which the Douglas E. Goldman Jewish Genealogy Center in Anu Museum has been granted consent or rights to publish online were used in the dataset generation. Moreover, as far as possible, all records containing living individuals have been removed from the dataset. Furthermore, all personal information and any information that can identify a specific person in this paper's examples, including the examples in the figures, have been altered to protect the individuals' privacy.

From the filtered GEDCOM files belonging to the above corpus, and after removing some files with parsing or encoding errors, three datasets were generated: Gen-SQuAD$_0$ using zero relation degree (SP and its spouses) with 6,283,082 questions, Gen-SQuAD$_1$ using first-degree relations with 28,778,947 questions, and Gen-SQuAD$_2$ using second-degree relations with 75,281,088 questions. Although all generated datasets contain millions of examples, only 131,072 randomly selected questions were used from each dataset when training the Uncle-BERT models. These were enough for the models to converge. Therefore, the size of the dataset did not impact the training results.

Each dataset was split into a training set (60%), a test set (20%), and an evaluation set (20%). To better evaluate the success of the different question answering models, the 131,072 questions in each dataset were classified into twelve types. Examples of questions and their classification types are shown in Table 3. Each question may refer to the *SP's* relationship type (e.g., Emily's grandson or by the direct name of the relative,e.g., Grace) and target one type of ontological entity as an answer (date, place, name, relationship type). Questions were classified into types based on the template, if generated using the template-based method (e.g., templates using place attributes were classified as “place”, and date attributes as “date”), based on the WH question (e.g., When questions were classified as “date”, and Where as “place”), if generated using the WH generation algorithm, or as general information / named entity, if generated by the DNN model. Therefore, the information / named entity may also include the other types of questions. It is important to note that these questions are semantically similar to the open-domain questions in SQuAD [90, 91] datasets.

[16] https://dbs.anumuseum.org.il/skn/en/c6/e18493701

[17] https://www.gov.il/BlobFolder/legalinfo/data_security_regulation/en/PROTECTION%20OF%20PRIVACY%20REGULATIONS.pdf

[18] https://gdpr-info.eu/

Table 3

Question types.

Question type / objective	**Examples**	**Source**
Name	What is Emily's full name? What is Emily's last name?	Rule-based Rule-based
Date	When was Emily born? When did Emily get married?	Rule-based Rule-based
Place	Where was Emily buried? Where did Emily live?	Rule-based Rule-based
Information / named entity	Who was Emily's first boyfriend? Did Emily go to college?	DNN DNN
First-degree relation	Who was Emily's son? Who was Jonathan?	DNN, rule-based DNN
Second-degree relation	How many sisters did Emily have? How many brothers did Emily have?	Rule-based Rule-based
First-degree date	When was Emily's husband born? When was John born?	Rule-based Rule-based
First-degree place	Where was Emily's father born? Where was Alexander born?	Rule-based Rule-based
First-degree information / named entity	What was Emily's father's academic degree? What was Alexander's illness?	DNN DNN
Second-degree date	When did Emily's sister die? When did Yalma die?	Rule-based Rule-based
Second-degree place	Where was Emily's grandson born? Where was Grace born?	DNN Rule-based
Second-degree information / named entity	What was Emily's grandfather's rank in the military? Where was Tim's first internship as a lawyer?	DNN DNN

4.2. Uncle-BERT fine-tuning

For fine-tuning Uncle-BERT[19], the generated Gen-SQuAD training datasets were used. Each context in the Gen-SQuAD$_0$, Gen-SQuAD$_1$, and Gen-SQuAD$_2$ datasets was lowercased and tokenized using WordPiece [132].

[CLS] When was Grace Williams born? **[SEP]** Mia's daughter (Emily Brown) was born in 1961 in Poland. Emily Brown had one husband. Mia Brown was buried in Poland. Mia's daughter (Grace Williams) was born in 1983 in Herzliya. Mia's daughter (Grace Williams) was a female. Mia Brown died in Poland. John Smith is Emily's husband. John Smith married Emily Brown in Tel Aviv. Emily Brown and Grace Williams were Mia's children...**[CLS]**

Fig. 12. Uncle-BERT model input example.

Figure 12 presents the model's input, where the [CLS] tag, which stands for classifier token, is the beginning of the input, followed by the first part of the input - the question. The [SEP] tag, which stands for a separator, separates the first part of the input (i.e., a question) and the second part – the context. [CLS] at the end indicates the end of the input.

To evaluate the effect of the depth of the consanguinity scope on the model's accuracy, an Uncle-BERT model was trained for each of the three datasets: Uncle-BERT$_0$ using Gen-SQuAD$_0$, Uncle-BERT$_1$ using Gen-SQuAD$_1$, and Uncle-BERT$_2$ using Gen-SQuAD$_2$. All models were trained with the same hyperparameters, that are shown in Table 4.

Table 4

Uncle-BERT training hyperparameters.

Hyperparameters	**Value**
Max question tokens	64
Max sequence tokens	512
Max answer tokens	30
Doc stride	128
Batch size	8
Learning rate	3e-5
Train size	131,072
Epocs	20

Max question tokens is the maximum number of tokens to process from the question input; if the question input length was greater than the Max question tokens, it was trimmed. Max sequence tokens are the maximum tokens to process from the combined context and question inputs.

If the cumulative context and question length was longer than the Max sequence tokens hyperparameter value, the context was split into shorter sub-texts using a sliding window technique; the Doc stride represents the sliding window overlap size. For example, consider the following hyperparameters' values: the max sequence tokens hyperparameter is 25, the doc stride hyperparameter is 6, and the following training example: "[CLS] When was Matt Adler's father born? [SEP] Matt's father (Noah Adler) was born in 1950 in London, England. Matt's father (Noah Adler) was a male. Matt's brother (Joanne Adler) was a male. Matt Adler was born in 1975 in London, England. Matt's mother (Carol) was born in 1950. [CLS]"; the question of the training example contains 7 tokens, and 18 tokens are left for the context. Therefore, the context will be split into three training examples: 1) "[CLS] When was Matt Adler's father born? [SEP] Matt's father (Noah Adler) was born in 1950 in London, England. Matt's father (Noah Adler) was a male [CLS]" (i.e., tokens 1 to 18), 2) "[CLS] When was Matt Adler's father born? [SEP] father (Noah Adler) was a male. Matt's brother (Joanne Adler) was a male. Matt Adler was born in [CLS]" (i.e., tokens 12 to 30), 3) "[CLS] When was Matt Adler's father born? [SEP] a male. Matt Adler was born in 1975 in London, England. Matt's mother (Carol) was born in 1950. [CLS]" (i.e., tokens 24 to 42). The model will be trained with the same question on the three new examples; if the answer span does not exist in an example, it is considered unanswerable.

Max answer tokens is the maximum number of tokens that a generated answer can contain. Train size is the number of examples used from the dataset during the training cycle.

As is customary with the SQuAD benchmark, an F1 score was calculated to evaluate Uncle-BERT models:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Precision equals the fraction of correct tokens out of the retrieved tokens (i.e., words that exist in both the predicted and the expected answer), and recall equals the fraction of the correct tokens in the retrieved (predicted) answer out of the tokens in the expected

[19] A link to the code: https://github.com/omrivm/Uncle-BERT

answer. This metric allows measuring both exact and partial answers.

5. Results

To evaluate the accuracy of the proposed fine-tuned models, the Gen-SQuAD$_2$ dataset was used to represent a real-world use-case in which a user is investigating her genealogical roots with the genealogical scope of two relation degrees (generations)[20]. To compare the model's accuracy for each type of answer, an F1 score was calculated to evaluate every Uncle-BERT model (i.e., Uncle-BERT$_0$ trained on Gen-SQuAD$_0$, Uncle-BERT$_1$ trained on Gen-SQuAD$_1$, and Uncle-BERT$_2$ trained on Gen-SQuAD$_2$). An overall accuracy evaluation of the three models was performed by calculating the F1 score for a mix of random questions of all types.

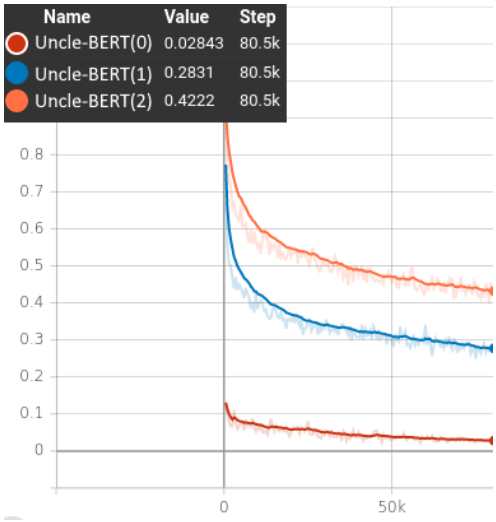

Fig. 13. The three Uncle-BERT model's train loss[21].

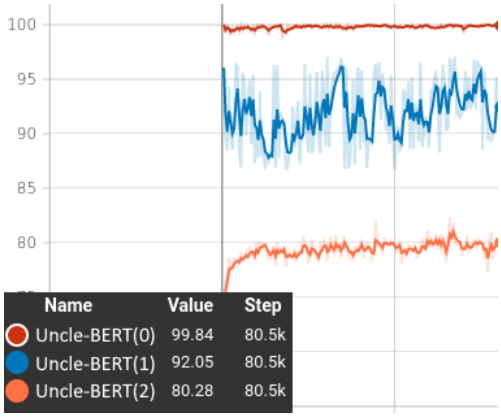

Fig. 14. The three Uncle-BERT model's train F1 score[22].

Figures 13 and 14 show the training loss and F1 scores of each of the three models. As expected, the more complex the context and questions, the lower the F1 score. While on narrow persons' contexts and questions (Gen-SQuAD$_0$), the model achieved an F1 score of 99.84; on second-degree genealogical relations (Gen-SQuAD$_2$), it achieved only an F1 score of 80.28.

Furthermore, as can be observed in Table 5, compared to the Uncle-BERT$_2$ model (trained with broader contexts of second-degree genealogical relations), the Uncle-BERT$_0$, which was trained using information about the *SP* and its spouses, fails to answer questions of any kind, including questions about the *SP* alone. We hypothesize that the model overfits to narrow contexts and therefore cannot handle larger context (Gen-SQuAD$_2$) "noise". This emphasizes the importance of the context size in the training data. Uncle-BERT$_1$ successfully answers most of the question types and even overtakes Uncle-BERT$_2$ in place-related questions. Except for place-related questions, it seems that a broader context improves the model's accuracy (Uncle-BERT$_2$).

Next, the best model, Uncle-BERT$_2$, was compared to several state-of-the-art open-domain question-answering DNN models. To this end, all the following models were trained using SQuAD 2.0 [90]: BERT [22], Distilbert [93], RoBERTa [70], Electra [18], DELFT [134]. Furthermore, to evaluate the effectiveness of the proposed genealogical question answering pipeline compared to the state-of-the-art knowledge graph-based pipeline, the genealogical adaptation of the DELFT model, Uncle-DELFT$_2$, was

[20] Similar to Anu Museum user interface - https://dbs.anumuseum.org.il/skn/en/c6/e8492037/Personalities/Weizmann_Chaim

[21] x-axis: number of epochs, y-axis: loss

[22] x-axis: number of epochs, y-axis: F1 score

created. Uncle-DELFT$_2$, based on BERT combined with the GNN graph traversal, was trained on Gen-SQuAD$_2$.

As can be observed in Table 6, the baseline BERT model trained on the open-domain SQuAD 2.0 achieved an F1 score of 83 on the open-domain SQuAD 2.0 dataset [90]. However, on the genealogical domain dataset (Gen-SQuAD$_2$), it achieved a significantly lower F1 score (60.12) compared to the Uncle-BERT$_2$ (81.45). The fact that Uncle-BERT$_2$ achieves a higher F1 score is not surprising since the model was trained on genealogical data, as opposed to the baseline BERT model trained on the open-domain question data. However, when comparing Uncle-BERT$_2$ to Uncle-DELFT$_2$, it is clear that the performance improvement is due to the proposed methodology and not just due to the richer or domain-specific training data. Moreover, the DELFT method is much more complex than BERT, yet it achieved a lower score even when trained on the same domain-specific data. The fact that the vast majority of entities (found in both the "user" question and the expected answer) exists only in the unstructured data makes it hard for the GNN to find the correct answer (i.e., to complete the graph). This finding emphasizes the uniqueness of a genealogical question answering task compared to the open-domain question-answering and the need for the end-to-end pipeline and methodology for training and using DNNs for this task, as presented in this paper. Since Uncle-BERT$_2$ achieved a higher accuracy score than the more complex Uncle-DELFT$_2$ model, we conclude that the proposed method reduces complexity while increasing accuracy.

As shown in Table 6, although some questions appear in both Gen-SQuAD$_2$ and SQuAD 2.0 datasets, there is still a significant difference between open-domain questions and genealogical questions. Except for Uncle-DELFT$_2$ in the case of date questions, all the state-of-the-art models failed to answer natural genealogical questions compared to Uncle-BERT$_2$ (and in many cases, even compared to Uncle-BERT$_1$). However, Uncle-DELFT$_2$ was successful regarding date questions. This may imply that objective date questions are harder to extract from unstructured texts and the graph structure contributes to resolving such questions. Moreover, BERT's success on SP's date questions (compared to Uncle-BERT$_2$) may suggest that these questions are more generic and have more common features among different domains than unique features in the genealogical domain. Furthermore, the current state-of-the-art knowledge graph pipeline (i.e., DELFT) achieved performance similar to simpler BERT-based models. This indicates that while it is beneficial for open-domain questions, it is not as effective in the genealogical domain. This result, combined with the additional complexity of DELFT, makes it less satisfactory in this domain (except for date questions, as mentioned above).

Interestingly, the "basic" BERT model outperforms all the newer BERT-based models (except for Uncle-BERT$_2$). Furthermore, the fact that Uncle-BERT$_1$ achieved a higher F1 score on place type questions may indicate that place type questions may be more sensitive to "noise" or broad context. For example, place names may have different variations for the same entity (high "noise"), e.g., NY, NYC, New York, and New York City are all references to the same entity. This variety makes the model's task more difficult, thus adding broader contextual information and other types of "noise" (e.g., other entities, more people names, and dates), which may reduce the model's accuracy. Another possible reason for Uncle-BERT$_2$'s lower accuracy on place type questions may be the fact that Uncle-BERT$_2$ was trained with both one-hop-away and two-hop-away contexts while Uncle-BERT$_1$ was trained only with one-hop-away contexts. The fact that the F1 score of the model is smaller on second-degree place objective questions (1.39) than on first-degree (4.72) and zero-degree (10.01) place objective questions may reinforce this indication. However, it is important to notice that in many cases, this factor will not affect the F1 score since the F1 score does not use the position of the answer (start and end index), but only the selected tokens compared to the answer tokens. Since most children and parents live in the same place, either the parent's place (e.g., birthplace) or the child's place can be selected by the model without affecting the F1 score. Table 7 presents some examples of answer predictions for place objective questions by Uncle-BERT$_1$ and Uncle-BERT$_2$. These results suggest that higher accuracy can be achieved by classifying the question types and using a different model for different question types and relation depths.

Table 5

Uncle-BERT models F1 score on Gen-SQuAD$_2$.

Question objective	**Uncle-BERT$_0$**	**Uncle-BERT$_1$**	**Uncle-BERT$_2$**
Name	44.53	95.14	**97.64**
Date	21.60	52.48	**55.10**
Place	27.54	**88.53**	78.52
Information \ named entity	16.91	15.22	**87.40**
First-degree relation	19.58	86.94	**89.45**
Second-degree relation	20.66	63.45	**82.52**
First-degree date	13.26	43.44	**53.85**
First-degree place	34.17	**86.55**	81.83
First-degree information / named entity	8.95	12.21	**87.28**
Second-degree date	11.68	43.12	**44.87**
Second-degree place	33.10	**80.51**	79.12
Second-degree information / named entity	8.37	11.34	**81.04**
Overall	19.73	69.92	**81.45**

Table 6

F1 scores of Uncle-BERT$_2$ and other state-of-the-art models on Gen-SQuAD$_2$.

Question objective	**BERT**	**Distilbert**	**RoBERTa**	**Electra**	**DELFT**	**Uncle-DELFT$_2$**	**Uncle-BERT$_2$**
Name	28.27	28.54	38.97	21.30	32.99	39.84	**97.64**
Date	60.58	53.30	44.33	34.92	39.62	**79.35**	55.10
Place	74.96	64.67	40.41	26.03	36.27	66.66	**78.52**
Information / named entity	71.20	66.79	34.96	31.72	40.91	70.58	**87.40**
First-degree relation	65.20	62.41	55.32	49.10	34.42	46.48	**89.45**
Second-degree relation	55.85	46.31	42.03	43.09	37.56	41.01	**82.52**
First-degree date	46.45	48.16	42.54	37.45	40.58	**64.84**	53.85
First-degree place	74.58	66.73	47.02	21.50	36.64	75.78	**81.83**
First-degree information / named entity	60.57	64.54	35.95	35.30	37.59	68.78	**87.28**
Second-degree date	39.49	39.75	23.30	26.99	38.29	**60.15**	44.87
Second-degree place	69.49	66.28	41.34	22.47	36.60	66.40	**79.12**
Second-degree information / named entity	60.19	62.38	34.37	37.15	35.70	47.26	**81.04**
Overall	60.12	60.19	39.45	43.39	37.56	42.96	**81.45**

Table 7

Uncle-BERT$_1$'s and Uncle-BERT$_2$'s prediction examples

Question objective	**Question**	**Context (relevant parts)**	**Correct Answer**	**Uncle-BERT$_1$**	**Uncle-BERT$_2$**
Place	Where was John born?	… John was born in Poland in 1866 … John grew up in PL until he was…	Poland	in Poland	PL
Place	Where was John buried?	… John died and was buried in Germany during … Kate (John's daughter) was born in France…	Germany	Germany	France
First-degree place	Where did John's father get married?	… Matt (John's father) was born in Warsaw, Poland … Matt married Elain in Warsaw…	Warsaw	Warsaw	in Warsaw
First-degree place	Where was Matt killed?	… Matt died at home in Poland surrounded… his father (John's grandfather) was killed in Pruszkow in 1850…	Poland	Poland	Pruszkow
Second-degree place	Where did John's grandfather die?	… his father (John's grandfather) was killed in Pruszkow in 1850…	Pruszkow	in Pruszkow	Pruszkow

6. Conclusions and future work

This study proposed and implemented a multi-phase end-to-end methodology for DNN-based answering natural questions using transformers in the genealogical domain.

The presented methodology was evaluated on a large corpus of 3,140 family trees comprised of 1,847,224 different persons. The evaluation results show that a fine-tuned Uncle-BERT$_2$ model, trained on the genealogical dataset with second degree relationships, outperformed all the open-domain state-of-the-art models. This finding indicates that the genealogy domain is distinctive and requires a dedicated training dataset and fine-tuned DNN model. A comparison of the proposed knowledge-graph-to-text approach was also found to be superior to the direct knowledge graph-based models, such as DELFT, even after domain-adaptation, both in terms of accuracy and complexity. This study also examined the effect of the type of question on the accuracy of the question answering model. The date-related questions are different as they can be answered with greater accuracy directly from the knowledge graph and may have more generic features than other question types, while place-related questions are more sensitive to noise than other question types. In addition, the evaluation results of the three Uncle-BERT models showed that the consanguinity scope of graph traversal used for generating a training corpus influences the accuracy of the models.

In summary, this paper's contributions are: (1) a genealogical knowledge graph representation of GEDCOM standard; (2) a dedicated graph traversal algorithm adapted to interpret the meaning of the relationships in the genealogical data (Gen-BFS); (3) an automatically generated SQuAD-style genealogical training dataset (Gen-SQuAD); (4) an end-to-end question answering pipeline for the genealogical domain; and (5) a fine-tuned question-answering BERT-based model for the genealogical domain (Uncle-BERT).

Although the proposed end-to-end methodology was implemented and validated for the question answering task, it can be applied to other NLP downstream tasks in the genealogical domain, such as entity extraction, text classification, and summarization. Researchers can utilize the study's results to reduce the time, cost, and complexity and to improve accuracy in the genealogical domain NLP research.

Possible directions for future research may include: (1) investigating the tradeoff between rich context passage generation and increasing the Gen-BFS scope, (2) integration with DNC or GNNs for dynamic scoping, (3) finding a method for classifying question types, (4) investigating the contribution of each question type to the accuracy of the model, and developing a model selection or multi-model method for each question type, (5) investigating larger contexts (relation degrees) using models that can handle larger input (e.g., Longformer [58] or Reformer [9]), (6) extending the Gen-BFS algorithm to handle missing family relations by adding a knowledge graph completion step while traversing the graph, (7) investigating the influence of the order of verbalized sentences and especially the order of person reference types, (8) investigating an architecture that will rank family trees (embedding the entire graph [37]) based on similarity to the question [92]) and eliminate the need for the user to select a family tree, (9) investigating the impact of spelling mistakes and out-of-vocabulary words on the quality of the results, (10) and training other transformer models on genealogical data to further optimize question answering DNN models for the genealogical domain.

References

[1] Abujabal, A., Yahya, M., Riedewald, M., & Weikum, G. (2017, April). Automated template generation for question answering over knowledge graphs. In Proceedings of the 26th international conference on world wide web (pp. 1191-1200). doi:10.1145/3038912.3052583

[2] Artés, J.C., Conesa, J., & Mayol, E. (2012). Modeling Genealogical Domain - An Open Problem. KEOD.

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In The semantic web (pp. 722-735). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-76298-0_52

[4] Ball, R. (2017). Visualizing genealogy through a family-centric perspective. Information Visualization, 16(1), 74-89. doi:10.1177/1473871615621592

[5] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., ... & Schneider, N. (2013, August). Abstract meaning representation for sembanking. In Proceedings of the 7th linguistic annotation workshop and interoperability with discourse (pp. 178-186).

[6] Barratt, Nick. "From memory to digital record." Records management journal (2009). doi:10.1108/09565690910937209

[7] Barzilai, & Borwein, J. M. (1988). Two-Point Step Size Gradient Methods. IMA Journal of Numerical Analysis. doi:10.1093/imanum/8.1.141

[8] Bastos, A., Nadgeri, A., Singh, K., Mulang, I. O., Shekarpour, S., Hoffart, J., & Kaul, M. (2021, April). RECON: Relation Extraction using Knowledge Graph Context in a Graph Neural Network. In Proceedings of the Web Conference 2021 (pp. 1673-1685). doi:10.1145/3442381.3449917

[9] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long document transformer. arXiv pre-print arXiv:2004.05150.

[10] Bezerianos, A., Dragicevic, P., Fekete, J. D., Bae, J., & Watson, B. (2010). Geneaquilts: A system for exploring large genealogies. IEEE Transactions on Visualization and Computer Graphics, 16(6), 1073-1081.

[11] Białecki, A., Muir, R., Ingersoll, G., & Imagination, L. (2012, August). Apache lucene 4. In SIGIR 2012 workshop on open source information retrieval (p. 17).

[12] Blackstone, William (1750). An essay on collateral consanguinity, its limits, extent, and duration, in Tracts chiefly relating to the antiquities and laws of England. Oxford: Clarendon Press, (3rd edition: 1771).

[13] Bordes, A., Chopra, S., & Weston, J. (2014). Question Answering with Subgraph Embeddings. EMNLP. doi:10.3115/v1/D14-1067

[14] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Advances in neural information processing systems (pp. 2787-2795).

[15] Cai, L., Zhou, S., Yan, X., & Yuan, R. (2019). A stacked BiLSTM neural network based on coattention mechanism for question answering. Computational intelligence and neuroscience, 2019. doi:10.1155/2019/9543490

[16] Chen, Y., Wu, L., & Zaki, M. J. (2020). Toward subgraph guided knowledge graph question generation with graph neural networks. arXiv preprint arXiv:2004.06015.

[17] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. doi:10.3115/v1/D14-1179

[18] Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2019, September). ELECTRA: Pretraining Text Encoders as Discriminators Rather Than Generators. In International Conference on Learning Representations.

[19] Cui, W., Xiao, Y., Wang, H., Song, Y., Hwang, S., & Wang, W. (2017). KBQA: Learning question answering over QA corpora and knowledge bases. Proceedings of the VLDB Endowment, 10(5), 565. doi:10.14778/3055540.3055549

[20] Dai, H. J., Lee, Y. Q., Nekkantti, C., & Jonnagaddala, J. (2020). Family History Information Extraction with Neural Attention and an Enhanced Relation-Side Scheme: Algorithm Development and Validation. JMIR Medical Informatics, 8(12), e21750. doi:10.2196/21750

[21] Deemter, K. V., Theune, M., & Krahmer, E. (2005). Real versus template-based natural language generation: A false opposition?. Computational linguistics, 31(1), 15-24. doi:10.1162/0891201053630291

[22] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).

[23] Dijkstra, E. W. (1959). A note on two problems in connexon with graphs. Numerische mathematik, 1(1), 269-271.

[24] Dong, L., Wei, F., Zhou, M., & Xu, K. (2015, July). Question answering over freebase with multi-column convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 260-269). doi:10.3115/v1/P15-1026

[25] Du, X., Shao, J., & Cardie, C. (2017, July). Learning to Ask: Neural Question Generation for Reading Comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1342-1352). doi:10.18653/v1/P17-1123

[26] Duboue, P. A., & McKeown, K. (2003). Statistical acquisition of content selection rules for natural language generation. doi:10.3115/1119355.1119371

[27] Embley, D. W., Liddle, S. W., Eastmond, T. S., Lonsdale, D. W., Price, J. P., & Woodfield, S. N. (2017). Conceptual modeling in accelerating information ingest into family tree. In Conceptual Modeling Perspectives (pp. 69-84). Springer, Cham. doi:10.1007/978-3-319-67271-7_6

[28] Fader, A., Zettlemoyer, L., & Etzioni, O. (2013, August). Paraphrase-driven learning for open question answering. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1608-1618).

[29] Fang, Y., Sun, S., Gan, Z., Pillai, R., Wang, S., & Liu, J. (2020, November). Hierarchical Graph Network for Multi-hop Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 8823-8838).

[30] Feldman, Y., & El-Yaniv, R. (2019, July). Multi-Hop Paragraph Retrieval for Open-Domain Question Answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2296-2309). doi:10.18653/v1/P19-1222

[31] Francisco, M.P., & Pérez, A.B. (2010). Projecte de recerca bàsica o aplicada PAC3 — Tercera Prova d'avaluació continuada.

[32] Galkin, M., Trivedi, P., Maheshwari, G., Usbeck, R., & Lehmann, J. (2020, November). Message Passing for Hyper-Relational Knowledge Graphs. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 7346-7359). doi:10.18653/v1/2020.emnlp-main.596

[33] Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017, July). Creating training corpora for nlg micro-planning. In 55th annual meeting of the Association for Computational Linguistics (ACL).

[34] Garg, S., Vu, T., & Moschitti, A. (2020, April). Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In Proceedings of the AAAI

Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 7780-7788). doi:10.1609/aaai.v34i05.6282
[35] Gellatly, C. (2009). Trends in population sex ratios may be explained by changes in the frequencies of polymorphic alleles of a sex ratio gene. Evolutionary biology, 36(2), 190-200. doi:10.1007/s11692-008-9046-3
[36] Gellatly, C. (2015). Reconstructing historical populations from genealogical data files. In Population recon-struction (pp. 111-128). Springer, Cham. doi:10.1007/978-3-319-19884-2_6
[37] Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78-94. doi:10.1016/j.knosys.2018.03.022
[38] Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., ... & Badia, A. P. (2016). Hybrid computing using a neural network with dynamic external memory. Nature, 538(7626), 471-476. doi:10.1038/nature20101
[39] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864). doi:10.1145/2939672.2939754
[40] Han, J., Cheng, B., & Wang, X. (2020, November). Open Domain Question Answering based on Text Enhanced Knowledge Graph with Hyperedge Infusion. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (pp. 1475-1481). doi:10.18653/v1/2020.findings-emnlp.133
[41] Harrah, D. (2002). The logic of questions. In Handbook of philosophical logic (pp. 1-60). Springer, Dordrecht. doi:10.1007/978-94-009-6259-0_12
[42] Harviainen, J. T., & Björk, B. C. (2018). Genealogy, GEDCOM, and popularity implications. Informaatiotutkimus. doi:10.23978/inf.76066
[43] He, B., Zhou, D., Xiao, J., Jiang, X., Liu, Q., Yuan, N. J., & Xu, T. (2020, November). Integrating Graph Contextualized Knowledge into Pre-trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (pp. 2281-2290). doi:10.18653/v1/2020.findings-emnlp.207
[44] Heilman, M., & Smith, N. A. (2010, June). Good question! statistical ranking for question generation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 609-617).
[45] Herskovitz, A. (2012). A Suggested Taxonomy of Genealogy as a Multidisciplinary Academic Research Field. Journal of Multidisciplinary Research (1947-2900), 4(3).
[46] Hey, D. (Ed.). (2010). The Oxford companion to family and local history. OUP Oxford.
[47] Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: the view from here. natural language engineering, 7(4), 275.
[48] Hochreiter, S., & Schmidhuber, J. J. (1997). Long short-term memory. Neural Computation, 9(8), 1–32. doi:10.1162/neco.1997.9.8.1735
[49] Hosking, T., & Riedel, S. (2019, June). Evaluating Rewards for Question Generation Models. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 2278-2283). doi:10.18653/v1/N19-1237
[50] Indurthi, S. R., Raghu, D., Khapra, M. M., & Joshi, S. (2017, April). Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers (pp. 376-385).
[51] Irshad, H., Montaser-Kouhsari, L., Waltz, G., Bucur, O., Nowak, J. A., Dong, F., ... & Beck, A. H. (2014). Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: evaluating experts, automated methods, and the crowd. In Pacific symposium on biocomputing Co-chairs (pp. 294-305). doi:10.1142/9789814644730_0029
[52] Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2021). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. IEEE Transactions on Neural Networks and Learning Systems. doi:10.1109/TNNLS.2021.3070843
[53] Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. IEEE Transactions on Big Data. doi:10.1109/TBDATA.2019.2921572
[54] Jones, K. S., & Van Rijsbergen, C. J. (1976). Information retrieval test collections. Journal of documentation.
[55] Kim, J., & Mooney, R. (2010, August). Generative alignment and semantic parsing for learning from ambiguous supervision. In Coling 2010: Posters (pp. 543-551).
[56] Kim, N. W., Card, S. K., & Heer, J. (2010, May). Tracing genealogical data with timenets. In Proceedings of the International Conference on Advanced Visual Interfaces (pp. 241-248). doi:10.1145/1842993.1843035
[57] Kim, Y. Convolutional Neural Networks for Sentence Classification. EMNLP (2014). doi:10.3115/v1/D14-1181
[58] Kitaev, N., Kaiser, L., & Levskaya, A. (2019, September). Reformer: The Efficient Transformer. In International Conference on Learning Representations.
[59] Klein, D., & Manning, C. D. (2002). Fast exact inference with a factored model for natural language parsing. Advances in neural information processing systems, 15, 3-10.
[60] Koch, I., Freitas, N., Ribeiro, C., Lopes, C. T., & da Silva, J. R. (2019, September). Knowledge Graph Implementation of Archival Descriptions Through CIDOC-CRM. In International Conference on Theory and Practice of Digital Libraries (pp. 99-106). Springer, Cham. doi:10.1007/978-3-030-30760-8_8
[61] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., ... & Toutanova, K. (2019). Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7, 453-466. doi:10.1162/tacl_a_00276
[62] Lai, T., Tran, Q. H., Bui, T., & Kihara, D. (2019, November). A Gated Self-attention Memory Network for Answer Selection. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 5953-5959). doi:10.18653/v1/D19-1610

[63] Lebret, R., Grangier, D., & Auli, M. (2016, November). Neural Text Generation from Structured Data with Application to the Biography Domain. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 1203-1213). doi:10.18653/v1/D16-1128

[64] Leskinen, P., & Hyvönen, E. (2020). Reconciling and Using Historical Person Registers as Linked Open Data in the AcademySampo Knowledge Graph.

[65] Levy, R., & Andrew, G. (2006, May). Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In LREC (pp. 2231-2234).

[66] Li, W., Peng, R., Wang, Y., & Yan, Z. (2020). Knowledge graph based natural language generation with adapted pointer-generator networks. Neurocomputing, 382, 174-187. doi:10.1016/j.neucom.2019.11.079

[67] Lin, C. Y. (2008, September). Automatic question generation from queries. In Workshop on the question generation shared task (pp. 156-164).

[68] Liu, T., Wang, K., Sha, L., Chang, B., & Sui, Z. (2018, April). Table-to-Text Generation by Structure-Aware Seq2seq Learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[69] Liu, X., Zhu, Y., & Ji, S. (2020, August). Web Log Analysis in Genealogy System. In 2020 IEEE International Conference on Knowledge Graph (ICKG) (pp. 536-543). IEEE. doi:10.1109/ICBK50248.2020.00081

[70] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[71] Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development, 1(4), 309-317. doi:10.1147/rd.14.0309

[72] Lukovnikov, D., Fischer, A., Lehmann, J., & Auer, S. (2017, April). Neural network-based question answering over knowledge graphs on word and character level. In Proceedings of the 26th international conference on World Wide Web (pp. 1211-1220). doi:10.1145/3038912.3052675

[73] Lussier, A. A., & Keinan, A. (2018). Crowdsourced genealogies and genomes. Science, 360(6385), 153-154. doi:10.1126/science.aat2634

[74] Ma, X., Zhu, Q., Zhou, Y., Li, X., & Wu, D. (2020). Asking Complex Questions with Multi-hop Answer-focused Reasoning. arXiv preprint arXiv:2009.07402.

[75] McGuffin, M. J., & Balakrishnan, R. (2005, October). Interactive visualization of genealogical graphs. In IEEE Symposium on Information Visualization, 2005. INFOVIS 2005. (pp. 16-23). IEEE. doi:10.1109/INFVIS.2005.1532124

[76] McRoy, S. W., Channarukul, S., & Ali, S. S. (2003). An augmented template-based approach to text realization. Natural Language Engineering, 9(4), 381. doi:10.1017/S1351324903003188

[77] 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).

[78] Mitra, A., & Baral, C. (2016, February). Addressing a Question Answering Challenge by Combining Statistical Methods with Inductive Rule Learning and Reasoning. In AAAI (pp. 2779-2785).

[79] Moussallem, D., Gnaneshwar, D., Ferreira, T. C., & Ngomo, A. C. N. (2020, November). NABU–Multilingual Graph-Based Neural RDF Verbalizer. In International Semantic Web Conference (pp. 420-437). Springer, Cham. doi:10.1007/978-3-030-62419-4_24

[80] Mulang, I. O., Singh, K., & Orlandi, F. (2017, September). Matching natural language relations to knowledge graph properties for question answering. In Proceedings of the 13th International Conference on Semantic Systems (pp. 89-96). doi:10.1145/3132218.3132229

[81] Mulang, I.O., Singh, K., Prabhu, C., Nadgeri, A., Hoffart, J., & Lehmann, J. (2020). Evaluating the Impact of Knowledge Graph Context on Entity Disambiguation Models. Proceedings of the 29th ACM International Conference on Information & Knowledge Management. doi:10.1145/3340531.3412159

[82] Nadgeri, A., Bastos, A., Singh, K., Mulang', I. O., Hoffart, J., Shekarpour, S., & Saraswat, V. (2021, Augustus). KGPool: Dynamic Knowledge Graph Context Selection for Relation Extraction. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 535–548. doi:10.18653/v1/2021.findings-acl.48

[83] Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, 193, 217-250. doi:10.1016/j.artint.2012.07.001

[84] Otterstrom, S. M., & Bunker, B. E. (2013). Genealogy, migration, and the intertwined geographies of personal pasts. Annals of the Association of American Geographers, 103(3), 544-569. doi:10.1080/00045608.2012.700607

[85] Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., & Pintscher, L. (2016, April). From freebase to wikidata: The great migration. In Proceedings of the 25th international conference on world wide web (pp. 1419-1428). doi:10.1145/2872427.2874809

[86] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543). doi:10.3115/v1/D14-1162

[87] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 2227-2237). doi:10.18653/v1/N18-1202

[88] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[89] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[90] Rajpurkar, P., Jia, R., & Liang, P. (2018, July). Know What You Don't Know: Unanswerable Questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 784-789). doi:10.18653/v1/P18-2124

[91] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016, November). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 2383-2392). doi:10.18653/v1/D16-1264

[92] Reimers, N., & Gurevych, I. (2019, November). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3982-3992). doi:10.18653/v1/D19-1410

[93] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[94] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61-80. doi:10.1109/TNN.2008.2005605

[95] Schaffer, S., Gustke, O., Oldemeier, J., & Reithinger, N. (2018). Towards chatbots in the museum. In mobileCH@ Mobile HCI.

[96] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681. doi:10.1109/78.650093

[97] Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603.

[98] Seyler, D., Yahya, M., & Berberich, K. (2015, May). Generating quiz questions from knowledge graphs. In Proceedings of the 24th International Conference on World Wide Web (pp. 113-114). doi:10.1145/2740908.2742722

[99] Sha, L., Mou, L., Liu, T., Poupart, P., Li, S., Chang, B., & Sui, Z. (2018, April). Order-Planning Neural Text Generation From Structured Data. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[100] Shandler, J. (2020). The Savior and the Survivor: Virtual Afterlives in New Media. Jewish Film & New Media, 8(1), 23-47. doi:10.13110/jewifilmnewmedi.8.1.0023

[101] Shao, L., Duan, Y., Sun, X., Gao, H., Zhu, D., & Miao, W. (2017, July). Answering Who/When, What, How, Why through Constructing Data Graph, Information Graph, Knowledge Graph and Wisdom Graph. In SEKE (pp. 1-6). doi:10.18293/SEKE2017-079

[102] Shao, T., Guo, Y., Chen, H., & Hao, Z. (2019). Transformer-based neural network for answer selection in question answering. IEEE Access, 7, 26146-26156. doi:10.1109/ACCESS.2019.2900753

[103] Sheng, S., Zhou, P., & Wu, X. (2019, November). CEPV: A Tree Structure Information Extraction and Visualization Tool for Big Knowledge Graph. In 2019 IEEE International Conference on Big Knowledge (ICBK) (pp. 221-228). IEEE. doi:10.1109/ICBK.2019.00037

[104] Singhal, A. (2001). Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4), 35-43.

[105] Smolenyak, M., & Turner, A. (2004). Trace your roots with DNA: Using genetic tests to explore your family tree. Rodale.

[106] Song, L., Zhang, Y., Wang, Z., & Gildea, D. (2018, July). A Graph-to-Sequence Model for AMR-to-Text Generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1616-1626). doi:10.18653/v1/P18-1150

[107] Speer, R., Chin, J., & Havasi, C. (2017, February). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).

[108] Stenzhorn, H. (2002). Xtragena natural language generation system using xml-and java-technologies. In COLING-02: The 2nd Workshop on NLP and XML (NLPXML-2002).

[109] Suissa, O., Elmalech, A., & Zhitomirsky-Geffet, M. (2019). Toward the optimized crowdsourcing strategy for OCR post-correction. Aslib Journal of Information Management. Vol. 72 No. 2, pp. 179-197. doi:10.1108/AJIM-07-2019-0189.

[110] Suissa, O., Elmalech, A., & Zhitomirsky‐Geffet, M. (2021). Text analysis using deep neural networks in digital humanities and information science. Journal of the Association for Information Science and Technology. doi:10.1002/asi.24544

[111] Sumner, J. L., Farris, E. M., & Holman, M. R. (2020). Crowdsourcing reliable local data. Political Analysis, 28(2), 244-262. doi:10.1017/pan.2019.32

[112] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. International Journal of Computer Vision.

[113] Tuominen, J., Hyvönen, E., & Leskinen, P. (2017). Bio CRM: A Data Model for Representing Biographical Data for Prosopographical Research. BD.

[114] van den Berg, R., Kipf, T. N., & Welling, M. (2017). Graph Convolutional Matrix Completion. stat, 1050, 25.

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

[116] Wang, S., & Jiang, J. (2016). Machine comprehension using match-LSTM and answer pointer.(2017). In ICLR 2017: International Conference on Learning Representations, Toulon, France, April 24-26: Proceedings (pp. 1-15).

[117] Wang, S., Wei, Z., Fan, Z., Huang, Z., Sun, W., Zhang, Q., & Huang, X. J. (2020, November). PathQG: Neural Question Generation from Facts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 9066-9075). doi:10.18653/v1/2020.emnlp-main.729

[118] Wang, S., Yu, M., Guo, X., Wang, Z., Klinger, T., Zhang, W., Chang, S., Tesauro, G., Zhou, B., & Jiang, J. (2018). R3: Reinforced Ranker-Reader for Open-Domain Question Answering. AAAI.

[119] Wang, W., Yang, N., Wei, F., Chang, B., & Zhou, M. (2017, July). Gated self-matching networks for reading comprehension and question answering. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 189-198). doi:10.18653/v1/P17-1018

[120] Wang, Z., Mi, H., Hamza, W., & Florian, R. (2016). Multi-perspective context matching for machine comprehension. arXiv preprint arXiv:1612.04211.

[121] Warby, S. C., Wendt, S. L., Welinder, P., Munk, E. G., Carrillo, O., Sorensen, H. B., ... & Mignot, E. (2014). Sleep-spindle detection: crowdsourcing and evaluating performance of experts, non-experts and automated

methods. Nature methods, 11(4), 385. doi:10.1038/nmeth.2855
[122] Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Harvard University.
[123] White, M., & Caldwell, T. (1998). EXEMPLARS: A practical, extensible framework for dynamic text generation. In Natural Language Generation.
[124] Williams, R. R., Hunt, S. C., Heiss, G., Province, M. A., Bensen, J. T., Higgins, M., ... & Hopkins, P. N. (2001). Usefulness of cardiovascular family history data for population-based preventive medicine and medical research (the Health Family Tree Study and the NHLBI Family Heart Study). The American journal of cardiology, 87(2), 129-135. doi:10.1016/S0002-9149(00)01303-5
[125] Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45). doi:10.18653/v1/2020.emnlp-demos.6
[126] Wolinsky, H. (2006). Genetic genealogy goes global: Although useful in investigating ancestry, the application of genetics to traditional genealogy could be abused. EMBO reports, 7(11), 1072-1074. doi:10.1038/sj.embor.7400843
[127] Xiong, C., Zhong, V., & Socher, R. (2016). Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604.
[128] Xu, P., & Barbosa, D. (2019, Junie). Connecting Language and Knowledge with Heterogeneous Representations for Neural Relation Extraction. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 3201–3206. doi:10.18653/v1/N19-1323
[129] Yamada, I., Washio, K., Shindo, H., & Matsumoto, Y. (2019). Global entity disambiguation with pretrained contextualized embeddings of words and entities. arXiv preprint arXiv:1909.00426.
[130] Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., ... & Lin, J. (2019, June). End-to-End Open-Domain Question Answering with BERTserini. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations) (pp. 72-77).
[131] Yao, L., Mao, C., & Luo, Y. (2019, July). Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 7370-7377). doi:10.1609/aaai.v33i01.33017370
[132] Yonghui, W., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Macherey, K. (2016). Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.
[133] Zhang, Z., Wu, Y., Zhou, J., Duan, S., Zhao, H., & Wang, R. (2020). SG-Net: Syntax-Guided Machine Reading Comprehension. In AAAI (pp. 9636-9643). doi:10.1609/aaai.v34i05.6511
[134] Zhao, C., Xiong, C., Qian, X., & Boyd-Graber, J. (2020, April). Complex Factoid Question Answering with a Free-Text Knowledge Graph. In Proceedings of The Web Conference 2020 (pp. 1205-1216). doi:10.1145/3366423.3380197
[135] Zhao, Y., Ni, X., Ding, Y., & Ke, Q. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 3901-3910). doi:10.18653/v1/D18-1424
[136] Zheng, W., Yu, J. X., Zou, L., & Cheng, H. (2018). Question answering over knowledge graphs: question understanding via template decomposition. Proceedings of the VLDB Endowment, 11(11), 1373-1386. doi:10.14778/3236187.3236192
[137] Zhong, W., Tang, D., Duan, N., Zhou, M., Wang, J., & Yin, J. (2019, October). Improving question answering by commonsense-based pre-training. In CCF International Conference on Natural Language Processing and Chinese Computing (pp. 16-28). Springer, Cham. doi:10.1007/978-3-030-32233-5_2
[138] Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., & Chua, T. S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774.
[139] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision (pp. 19-27). doi:10.1109/ICCV.2015.11
[140] Zou, L., Huang, R., Wang, H., Yu, J. X., He, W., & Zhao, D. (2014, June). Natural language question answering over RDF: a graph data driven approach. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data (pp. 313-324). doi:10.1145/2588555.2610525
[141] Ganhotra, J., & Joshi, S. (2021). Does Dialog Length matter for Next Response Selection task? An Empirical Study. arXiv preprint arXiv:2101.09647.