Summary

This paper was written by Morten Jørgensen of Aalborg University for their masters thesis. This project has been part of a larger project called ”MEHDIE”. MEHDIE stands for ”Middle East Heritage Data Integration Endeavour” and their goal is to make it easier to ”integrate” data from different sources in the middle east. They have had earlier projects where they created models that used transliteration and phonetic similarities to find matches in the data and connect toponyms that represent the same real-world location. This papers contribution to the MEHDIE project by the creation of a number of Large Language Models. These models are able to take two toponyms as input and then output if they are a match or not.

BERT and Large Language Models are a very popular field of study, so a lot of research has already been conducted on the use of BERT models. But to the knowledge of the author not a lot of research has been done using BERT for toponyms matching of ancient/historic toponyms.

BERT models have two training phases, pre-training and fine-tuning. This means two sets of datasets have to used to successfully create a BERT model. Pre-training requires by far the largest dataset, while fine-tuning requires a much smaller dataset specific to the task the model should solve.

When working with BERT based models there are in general three different approaches. The first one is finding an already pre-trained model online, meaning you only have to pre-train it using your task specific dataset. The second case is that you pre-train a BERT model from scratch using your own data and then also fine-tune it. This is quite time consuming as pre-training a BERT model is very slow and lots of data is needed to create a well performing model. The third and last case is that you again find an already pre-trained model, but then you continue its pre-training for a bit using your own data to adapt it to your data domain, and then fine-tuning it. There are cases where all three cases are the correct choice, so in this project all three approaches were tried.

For the pre-training of models in this paper two datasets were combined and used. The first dataset is called ”Sefaria” and was created by the Sefaria project [1]. This dataset contains ancient manuscripts in Hebrew and their English translations. The other dataset used for pre-training is called ”roots_ar_openiti_proc” and was created by BigScience [2]. This dataset is a subset of the OpenITI dataset, which is a Machine-Readable Corpus of Islamicate Texts by Open Islamicate Texts Initiative [3].

For fine-tuning the models data from the MEHDIE project were used. This dataset consisted of data files with toponyms and relation files that indicate matches between toponyms. By using one of the models created in this paper we were able to expand the fine-tuning data by having a language expert look at the false positives produced by the model. This increased the number of matches in the fine-tuning dataset by about 61%.

Surprisingly the evaluation showed that using a pre-trained model called mBERT from Google in its cased version resulted in far better results, even compared to other pre-trained models like XLM-R from Facebook, which on paper generally outperforms mBERT in all task categories. We presented some theories to why this might be the case, but more research and work is needed to conclude why this happened. Some guesses included that a non-exhaustive test dataset were used for fine-tuning, which could lead to poorer performance as the models are punished by correct guess, which are marked as non-matches in the dataset.

[1] www.sefaria.org

[2] www.huggingface.co/bigscience-data

[3] www.openiti.org

Exploring the Efficacy of Specially-Trained Transformers on Geospatial Entity Matching of Historic Toponyms

Morten Jørgensen
Department of Computer Science
Aalborg University
Aalborg, Denmark
Email: mjarg19@student.aau.dk

Abstract—**Substantial effort has been put into digitizing and extracting information from historical and ancient manuscripts. These efforts often focus on a single civilization, its language, and culture. Thereby isolating these efforts and making it harder to collaborate and share knowledge between them. Some works have tried to connect these efforts and their data based on toponym matches using traditional methods such as transliteration for toponym matching. However, results have been uneven. The advent of transformer-based language models such as BERT has brought about improved performance in many language-related tasks, including toponym matching. However, these language models are often trained over large corpora of modern text in English. Even multilingual models are often trained on modern texts collected on the web. Here, we examine whether creating specially-trained multi-lingual models over ancient texts matching the toponym languages can be beneficial for this task. In this paper, we examine several methods using ancient manuscripts to adapt BERT-based models to identify matching toponyms in Arabic and Hebrew, two related Semitic languages with historical dialects and sizeable corpora of ancient texts. We evaluated our methods on a historical toponym matching task comprising several datasets of toponyms extracted from Middle East scholars The evaluation results were surprising in that the models presented in this work were outperformed by a multilingual model (mBERT) that was pre-trained on modern data.**

1. Introduction

Humans have used written languages for thousands of years to document events, recipes, stories, contracts, love letters, and much more. Not many ancient texts have survived the passage of time, with the vast majority being lost. Scholars are currently undertaking a number of projects to digitize and analyze these ancient texts to further our understanding of the ancient civilizations and humans that once were.

However, such projects typically only focus on a single civilization with its own written language. For example, the Syriaca project focuses on Syriac[1], the OpenITI project focuses on Islamic cultures[2] and the ”Beyond the Text” project focuses on Egypt during the Graeco-Roman period[3]. This focus creates isolated islands of information that a scholar or group of scholars are experts on but makes it hard to collaborate with other scholars who specialize in other civilizations. Because these fields of research are so separated, it is easy to think that the texts, events, and places that they research are also completely separate in time and space. But this is sometimes not the case as ancient civilizations interacted and exchanged ideas, stories, and more. This is why some endeavors [1] [2] have been started recently with the goal of uniting several cross-lingual toponym datasets to create a shared data map. However, these projects face a significant challenge as no automated tools are readily available for mapping between ancient place names in different languages and scriptures. Different approaches have been tried, such as transliteration [3] and Phonetic Similarity [4] and for modern toponyms Natural Language Processing (NLP) approaches [5] are widely used. Transliteration is where texts of different languages' alphabets are transformed into the same common alphabet. This method has the drawback of possible loss of meaning. A modern example of this loss of meaning could be the name of the Israeli city *Beersheba*, which in Arabic and Hebrew can be translated to ”Well of the seven.” But in English, it is just the name of a geographic location that has no deeper meaning. The transliteration has only made it possible for users of the Latin alphabet to read but not understand the name. Phonetic Similarity uses fairly similar principles, but instead uses phonetic similarities and distance between toponyms to deduce whether they match.

Over the last couple of years, there have been many developments in the area of NLP with new and more powerful models being introduced. One of the most popular NLP models is Bidirectional Encoder Representations from Transformers (BERT) [6]. Several works have shown that BERT models can perform at or close to the state-of-the-art in a wide area of NLP tasks [7]. In a recent survey [8] the authors examined seven different fields within NLP and their best-performing models. Six of the best-performing models

[1] www.syriaca.org/index.html

[2] www.openiti.org/about.html

[3] www.beyondthetext.ch/

incorporated a BERT model into their architecture. One of these is DocBERT [9], which is used in the field of `Text Classification`. By adding a fully connected layer to the end of BERT and then fine-tuning it they were able to achieve state-of-the-art results. They compared DocBERT to a number of classical NLP models over four datasets where *BERT-large* gave the best results followed by *BERT-base* in all datasets.

BERT models have a high textual understanding as a state-of-the-art model in the field of `Text Summarization` has been made using BERT to encode the input sequence into context representations [10]. The decoding was split into two phases, the first one uses a transformer decoder to create a draft output which is then masked and feed into BERT which is then feed into a decoder that then predicts each masked word, generating the summarized text. Using this approach they were able to achieve state-of-the-art performance of the CNN/Daily Mail datasets.

BERT also has a multilingual counterpart called Multilingual BERT (mBERT) which has been trained on 104 different languages. The amount of data used for training in each language differs quite a bit, as the Wikipedia dataset was used, which leads to different levels of performance between languages [11] [12] [13] [14]. Furthermore, several papers have found that monolingual BERT models perform better than mBERT on their target language [15] [16] [17].

Both BERT and mBERT are trained on modern texts like the Wikipedia dataset and the BooksCorpus dataset [6]. However, recent research shows that machine learning models perform better on historical texts when they are trained exclusively on historical texts instead of a mix between modern and historical texts [18].

In this paper, we explore if better ancient toponym entity matching results can be achieved by creating a specialized Ancient Semitic BERT (asBERT) model trained on ancient manuscripts compared to using a mBERT model trained on modern manuscripts and an extension of mBERT that has been pre-trained for a couple epochs on ancient manuscripts.

The contributions of this paper can be summarized as follows.

- BERT for ancient Arabic, ancient Hebrew, and a subset of English
- Extended mBERT for Arabic, Hebrew, and a subset of English
- Empirical evaluation on test sets to see which model completes the ancient toponym entity matching task the best

The rest of the paper is structured as follows. In Section 2 we review the use of BERT based model on ancient and historic text and on the task of toponym matching. Section 3 outlines preliminary information for this paper. In Section 4 the BERT models created for this paper is presented together with the datasets used to pre-train them. Then in Section 5 we present a number of experiments to evaluate the models. Finally we conclude in Section 6.

2. Related Work

The vast majority of BERT models are trained on modern web-based sources. However, several works have shown that for tasks involving ancient languages, training machine learning models on ancient variants of the task's target language improves performance with respect to models train on modern texts or a mix of ancient and modern texts [18]. This is unsurprising as substantial differences exist between ancient and modern texts in the same language [19].

However using a modern BERT as a starting point, can still be beneficial. Expanded Ancient Greek BERT [20] initially only pre-trained their model on ancient texts, but because of bad results they decided to use an already pre-trained modern Greek BERT model as a starting point for their own pre-training. This modern Greek BERT [21] was trained on the Greek Wikipedia, Greek Common Crawl, and the Greek part of the European Parliament Proceedings Parallel Corpus. By then, pre-training this model further with their ancient manuscripts, they were able to achieve a perplexity score of 4.9 on their test set and state-of-the-art performance on Part-of-Speech tagging. Perplexity score is a measure of the confidence and accuracy of a NLP models prediction, indicating its understanding of language and confidence in its guess. A perplexity score of 1.0 indicates a perfect model, while higher scores indicate worse performances.

There are also cases where only using ancient manuscripts for pre-training BERT has achieved the best results. For an ancient Chinese Automatic Word Segmentation and Part-of-Speech Tagging task [22], the authors used the pre-trained SIKU-RoBERTa model. The SIKU-RoBERTa model is a pre-trained Robustly optimized BERT approach (RoBERTa) [23] model that is trained on the "SiKuQuanShu" corpus, which is a collection of ancient Chinese texts. A RoBERTa model is an extension of BERT where the pre-training has been optimized and shown to perform better. The authors showed that the pre-trained SIKU-RoBERTa model performs better on the before-mentioned tasks than a number of LSTM based models that were also trained on ancient Chinese texts.

Based on the above-mentioned work, it seems that both mixed modern and ancient and only ancient BERT models can achieve state-of-the-art performance. Therefore, we will try both approaches in this paper to see which one performs the best for our task. The pre-trained model that will be used has to have been pre-trained on all target languages, while the model we will pre-train from scratch will be based on the ideas from RoBERTa. Previous works have focused on a single language and its ancient variant. In this work we will examine the feasibility for multiple, related ancient languages and if proficient BERT models for these languages can be created.

BERT models have been shown to perform well in toponym-based tasks, such as matching, identification, and recognition. TIMBERT [24] is a BERT model created to help epidemiologists better study the spread of viruses.

Some virus data-records lack geospatial metadata making it harder to track the spread of viruses. Using a pre-trained BERT model they were able to very accurately identify toponyms in medical articles and thereby expand the data-records. Another example of BERT's usage in this field is TopoBERT [5]. TopoBERT was created as a "plug and play" toponymn recognition model for social media and news media data. Here they used a pre-trained BERT model such as `bert-base-cased` or `bert-large-cased` for analyzing the inputs and then added a one-dimensional Convolutional neural network (CNN) for classification. These two works show how good BERT is at extracting information from text and then using BERT's output for classification in two different fields of work. Other works have also shown that BERT's output can be combined with other channels of information to enhance performance further. Geo-ER [25] seeks to make Entity Resolution (ER) less daunting by relying less on human-made rules. Geo-ER consists of three main components. The first is a BERT based model that analyses textual attributes. The second is a *Distance Embedding* component that computes the distances between two entities. The third and final component is called *Neighbourhood Attention* and is tasked with embedding the information of the surrounding entities. The output of all three components is then given as input to a fully-connected layer, which then makes a prediction. Using this model, they were able to outperform state-of-the-art models on a dataset comprising data from real-world location-based services like Yelp.

In this work, we will focus on the matching of ancient toponyms coming from historical sources. Here we will take inspiration from Geo-ER and built a model that is able to use multiples types of information available to us in these historical sources to enhance performance of the model.

Not many works have been concerned with the matching of ancient toponyms, but one of the few examples [4] utilizes phonetic similarity between toponyms to match them. The authors created two methods for toponym matching, one using the before-mentioned phonetic similarities between Hebrew and Arabic and the other using direct transliteration. Usually, transliteration uses an intermediate alphabet, such as Latin letters, for comparison. However, the authors proposed transliterating Hebrew directly into Arabic and vice versa in their method. Their evaluation showed that both methods performed better than traditional transliteration into Latin letters. Furthermore, they found that the best results would be achieved if both methods were used in combination. A reason for this is that the regions from which the datasets originated are different, which leads to a difference in pronunciation, affecting the phonetic method. The transliteration method was not affected by this and, therefore, performed better in these cases. The place names in the five datasets provided by this paper are written in ancient/medieval Hebrew and Arabic, which the authors used to benchmark their methods.

In this work, we will extend their work by attempting to use a BERT-based model trained on a multilingual corpus of ancient Hebrew and Arabic, which will be evaluated on the same datasets they used for their bench marking.

3. Preliminaries

In this section, the problem of toponym matching for this project is first formalized, and then some background information about BERT models is presented.

3.1. Toponym matching

In the simplest terms toponym matching refers to the task of matching names of real-world locations that refer to the same real-world location. More formally it can be stated as having two datasets D_1 and D_2 where the aim is to find all pairs of entities (e_i, e_j) where e_i is from D_1 and e_j is from D_2 that refer to the same real-world location, which we will call a `Match`. Each entity e has a set of attributes describing it, textual information $\{name, variants, description\}$, spatial position $\{longitude, latitude\}$ and temporal position $\{time\_start, time\_end\}$.

In this work we aim to create a `Matching` model that given two datasets D_1 and D_2 is able to accurately classify candidate matches (e_i, e_j) as either a `Match` or `Non-Match`.

3.2. BERT

BERT [6] was introduced in 2018 by a number of AI researches at Google. BERT is a specialization of the Transformer model [26] which was introduced in 2017. In simple terms a Transformer consists of two parts a encoder and a decoder. BERT is built as a stack of these encoders on top of each-other feeding into each other. A visualization of this can be seen in Figure 1. After passing the input data through all the encoders a number of embeddings equal the amount of input tokens is returned. These embeddings have been found to very accurately represent the textual and semantic information of the input sequences, making BERT able to perform well in many NLP tasks.

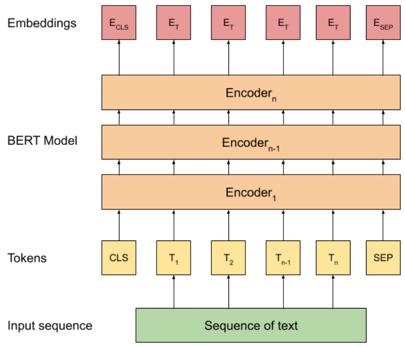

Figure 1. Overview of inputs and outputs of a BERT model. CLS and SEP denote special tokens that indicate beginning of sequence and end of sequence respectively.

	BERT-base	BERT-large
Encoders	12	24
Self-Attention heads	12	16
Dimension of embeddings	768	1024
Total parameters	110 M	340 M

Table 1. An overview of the differences between BERT-base and BERT-large.

In generel BERT comes in two different sizes, BERT-base and BERT-large and their differences can be seen in Table 1.

3.2.1. Tokenizer.

Before a sequence of text can be given to a BERT model, it has to be tokenized by a tokenizer. The tokenizer's job is to turn human-readable text into tokens a BERT model can understand. An example of this could be: "My BERT model is really good at question answering," which could become something like this: [345, 123, 789, 451, 127, 834, 521, 185, 414]. Where each of these IDs encodes a word in the input sequence.

There exist many different tokenizers with different methods of tokenization, but the most common ones for BERT models are `WordPiece` used in the original BERT [6] and `Byte-Pair Encoding (BPE)` used in RoBERTa [23] models. These two tokenizers are very similar and their main difference lies in how they create their vocabulary. Both tokenizers start by creating a list of all individual symbols and their frequencies. How they then merge these symbols and add them to the vocabulary is different. BPE adds the most frequent symbol pairs to its vocabulary while `WordPiece` instead chooses the ones that maximize the likelihood of the training data. This can be seen written in probability terms in Equation 1 and 2.

$$BPE : P(A, B) \tag{1}$$

$$WordPiece : \frac{P(A, B)}{P(A) * P(B)} \tag{2}$$

3.2.2. Training BERT.

Training a BERT model consists of two steps, namely pre-training and fine-tuning.

The goal of the pre-training phase is to give the model a general understanding of language and text. This can be done through a number of different training procedures. The most common and the one used in this paper is called Masked Language Modelling (MLM). The basic concept is that given an input sequence we *Mask* a certain percentage of the sequence, which the BERT model then has to guess. An example of this is the following text sequence: "I love using my bike when the weather is nice" which could be masked to: "I love using my [MASK] when the weather is nice". Based on the guess by BERT the loss is back-propagated through the model. Pre-training is by far the most time-consuming part of creating a BERT model as corpora typically contain over a billion words, and the model is trained for many epochs to give it a general understanding of language. But because BERT models gain a general understanding of languages during pre-training, it can be reused for many different tasks. Therefore, it is often not necessary to pre-train your own model; instead, using an already pre-trained model, which you can then fine-tune, is often a better option.

Fine-tuning a BERT model is far less time-consuming as the training is only specific to the task it should do. For example, the BERT model should perform Sentiment analysis for user comments on a website. We first have to collect a fine-tuning dataset that contains comments and a label for each indicating the tone of the comment. Then, we can take a BERT model and add a fully connected layer to the end of it with a neuron for each unique label in our fine-tuned dataset. This model can then be trained for a number of epochs to improve its performance.

4. Methodology

In this section, the technical aspects of this paper will be reviewed. Firstly, the different approaches to utilizing a BERT model will be examined in Section 4.1. Then the datasets that were used to pre-train the BERT models will be presented in Section 4.2. In the following Sections after that, the BERT models will be presented with their configuration and training procedures explained.

4.1. Bert Models

There are three different approaches to creating a new BERT based model, these three approaches have been shown in Figure 2. Three steps are present in the Figure. Pre-train refers to the task of training a BERT model with a large corpus of data, such that it gains a general understanding of language structure and semantics. Because BERT models gain a general understanding during this step, BERT models can be reused for many different tasks. This is the case for row a) in Figure 2 with mBERT as it has been pre-trained by Google on data from 104 different languages. More precisely, we are referring to the cased version called cased-mBERT, but we will refer to it just as mBERT. This model can be adapted during the Fine-tune step using the datasets described in Section 5.2 to make it perform toponym matching.

Row *b)* in Figure 2 shows the second method, which is a lot more time-consuming as a BERT model has to be pre-trained with the datasets from Section 4.2 and then fine-tuned on the task data from Section 5.2. This will be explained more in Section 4.4, where we will present our own model asBERT, which follows this approach.

Finally, there is the third method shown as row *c)* in Figure 2. This approach is a combination of the previous two where we take a pre-trained model like mBERT and then extend its pre-training by a couple of epochs with our domain-specific data from Section 4.2. This will explained

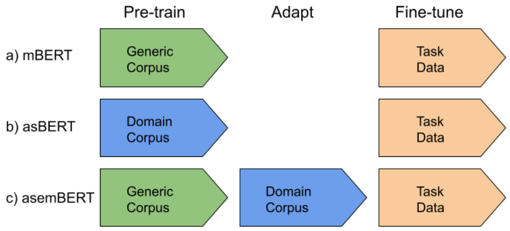

Figure 2. The three alternatives when deciding which BERT model to use.

	Words	Tokens
OpenITI	1.41B	1.68B
Sefaria	0.44B	0.57B
Total	1.85B	2.25B

Table 2. Words indicate the total amount of words in the cleaned datasets, while Tokens show how many tokens were produced by running the tokenizer on the datasets.

in more detail in Section 4.3, where we present our model called Ancient Semitic Extended Multilingual BERT (asemBERT) that follows this approach.

4.2. Datasets

The two datasets used for pre-training the BERT models created in this paper will be described in the following section. The first dataset is called ”Sefaria” and was created by the Sefaria project[4]. The dataset contains ancient manuscripts in Hebrew and their English translations. The other dataset used for pre-training is called ”roots_ar_openiti_proc” and was created by BigScience[5]. The dataset is a subset of the OpenITI dataset, which is a Machine-Readable Corpus of Islamicate Texts by Open Islamicate Texts Initiative [6]. From now on, the dataset” roots_ar_openiti_proc” will be referred to as ”OpenITI” for simplicity’s sake. The sizes of the datasets after removing unwanted characters and numbers can be seen in Table 2.

For calculating the perplexity of the BERT models after pre-training, a subset of the combined datasets will be used, which comprises 10% percent of the full dataset.

4.3. Ancient Semitic Extended mBERT

With asemBERT, we are extending an already pre-trained model as shown in Figure 2. In this paper, we choose *Google's* mBERT as it is capable of both Arabic and Hebrew. We then pre-trained it for 5 epochs on four NVIDIA A10 cards for two days using our domain-specific datasets described in Section 4.2. Before pre-training, mBERT had a perplexity of 19.66 on the test part of the dataset. But after pre-training and turning mBERT into asemBERT the perplexity had dropped all the way down to 2.79. This drop indicate that the model has learned and adapted to the dataset and increased its understanding of ancient Hebrew and Arabic.

4.4. Ancient Semitic BERT

In the original BERT paper [6], they created two different sizes of BERT, `base` and `large`. In general, BERT-`large` performs a bit better [27], but it comes at the cost of the model having three times as many parameters, slowing down training and inference times. For practical reasons, we opted to use the same size for asBERT as BERT-`base` because of the much longer training times when using BERT-*large*.

Tokenizer. For asBERT the BPE tokenizer was chosen. It was chosen based on the findings from RoBERTa [23], where they got better results using BPE compared to the WordPiece tokenizer used in the original BERT [6]. The vocabulary size of the tokenizer was set to `52000` because RoBERTa also increased their vocabulary size from the original BERT’s `30522`.

Pre-Training. asBERT was pre-trained on the two datasets described in Section 4.2 for 40 epochs. It took 14 days on four NVIDIA A10 cards. When deciding on training parameters, a lot of inspiration was taken from the research of RoBERTa [23]. In the original BERT [6], they thought that the Next Sentence Prediction (NSP) task was an important part of the pre-training procedure for BERT. But with RoBERTa the authors found that in some cases removing NSP and only using having MLM as the pre-training procedure matches or slightly improves downstream task performance. Because of this, we decided to only use MLM for the pre-training of asBERT. At the end of pre-training, asBERT had achieved a perplexity score of 6.77 on the test subset of the dataset. Giving it a worse perplexity than asemBERT, which could indicate that using mBERT as a starting point gives a basic understanding of languages and therefore better results. asBERT did get quite a lower perplexity score than mBERT (asemBERT before pre-training), which is to be expected as mBERT has never seen ancient Arabic or Hebrew before.

5. Empirical Evaluation

In this section we will evaluate the performance of mBERT, asemBERT and asBERT on a cross-lingual toponym matching task. To do this we will use the evaluation strategy presented in Section 5.1. Then the task-specific datasets used for evaluation will be presented in Section 5.2. After that in Section 5.3 the evaluation setup will be presented where the metric used for evaluation is presented. Then the results of the experiments will be presented in Section 5.4. Lastly, a discussion of the results from the experiments will be conducted in Section 5.5.

[4] www.sefaria.org
[5] www.huggingface.co/bigscience-data
[6] www.openiti.org/

5.1. Evaluation Strategy

The three BERT based models will be tested in a number of experiments and the overall architecture used for these experiments can be seen in Figure 3. The evaluation model is inspirited by Geo-ER [28], but with the addition of a spatial component as some of the toponyms in the task-specific datasets contain this information. The BERT models will be tested in the full evaluation model with all components, but also in sub-models where some components have been disabled to see how this effects performance.

5.2. Toponym matching datasets

The five task-specific datasets presented in Phonetic Similarity for Cross-source and Cross-language Toponym Matching [4] will be used for the evaluation of the presented BERT models on the task of toponym matching. In Table 3 an overview of these datasets can be seen.

#	Dataset 1 (Entries)	Dataset 2 (Entries)	Old Matches	New Matches
1	YaqutSham (687)	KimaSham (1899)	33	56
2	ThurayaSham (291)	KimaSham (1899)	21	50
3	Tudela (306)	Althurayya (2241)	18	41
4	Yaqut (484)	Kima (Andalus/Magreb) (559)	33	33
5	Damast (447)	Tudela (306)	32	41
			137	221

Table 3. Table showing an overview of the task-specific datasets used for evaluation of the BERT based models.

Each toponym entry in the datasets has a number of attributes that describe the toponym. The datasets in Table 3 do not all have the same attributes and some are not relevant for our toponym matching task. Therefore, we decided to transform all the datasets into the same format with the same attributes. These are: *id*, *name*, *variants*, *description*, *time_start*, *time_end*, *longitude* and *latitude*. Not all dataset entries contain the relevant attributes, which means that some fields are left empty for some toponyms. There is also quite a big difference in the content of the field *description*. For *id* dam51 its *description* contains ”The modern town of al-Busayra in Syria”, while many others just contain ”human settlement” or multiple sentences of text. Because of this variety in quality of the *description* field, it will be tested in Section 5.4.1 whether this field should be included in the textual inputs at all. For all textual fields Arabic, Hebrew and English are all used interchangeably.

The list of matched toponyms between the datasets are not exhaustive. This became apparent when running a fine-tuned mBERT on the datasets and then having an expert look over the false positives. They were able to find 84 additional matches and thereby improving the quality of the datasets drastically. There is a chance that more unobserved matches are still present in the datasets. This could lead to misleading results as a model might find matches that are not part of the datasets and therefore get a lower evaluation score.

5.3. Evaluation Setup

All the experiments were conducted on the same machine which is equipped with a V100 NVIDIA graphics card. Each experiment is run for 20 epochs with a batch size of 32 and a learning rate of $3e-5$ with the AdamW optimizer and CrossEntropyLoss from *PyTorch*. For all experiments, cross-validation is used with 10 folds of the total dataset, where the 9 first folds are used for training while the last is used for testing. This was done 10 times, so each fold was used for testing once. Other evaluation settings were also tried, but lowering the folding factor or using early stopping for the training did not substantially change the results.

The metrics used in the evaluations are Precision, Recall, and F5 scores, which are presented below.

With the Precision equation in 3 we can answer what proportion of positive guesses were actually correct.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

With the Recall equation in 4 we can identify how big a proportion of all positives were identified correctly.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

With the F5 equation in 5 we emphasize Recall five times more than Precision. This is appropriate if emphasizing identifying all positives over how many of the actual guesses were correct is deemed more important.

$$F5 = (1 + 5) * \frac{Precision * Recall}{(5 * Precision) + Recall} \tag{5}$$

The F5 score will be used as the principal comparison metric for the results in Section 5.4, while Recall and Precision will be used to further our understanding of the results.

5.4. Results

In this section, the results of the evaluation will be presented. As several textual attributes are available, we will examine the effect of different amounts of textual input on the BERT model in the first experiment. Then, we will compare the different BERT models proposed in this paper. Thereafter, we will compare the performance of different pre-trained BERT models. Lastly, we will test the importance of the different component in the evaluation model, by removing one component for each test to see how it affects performance.

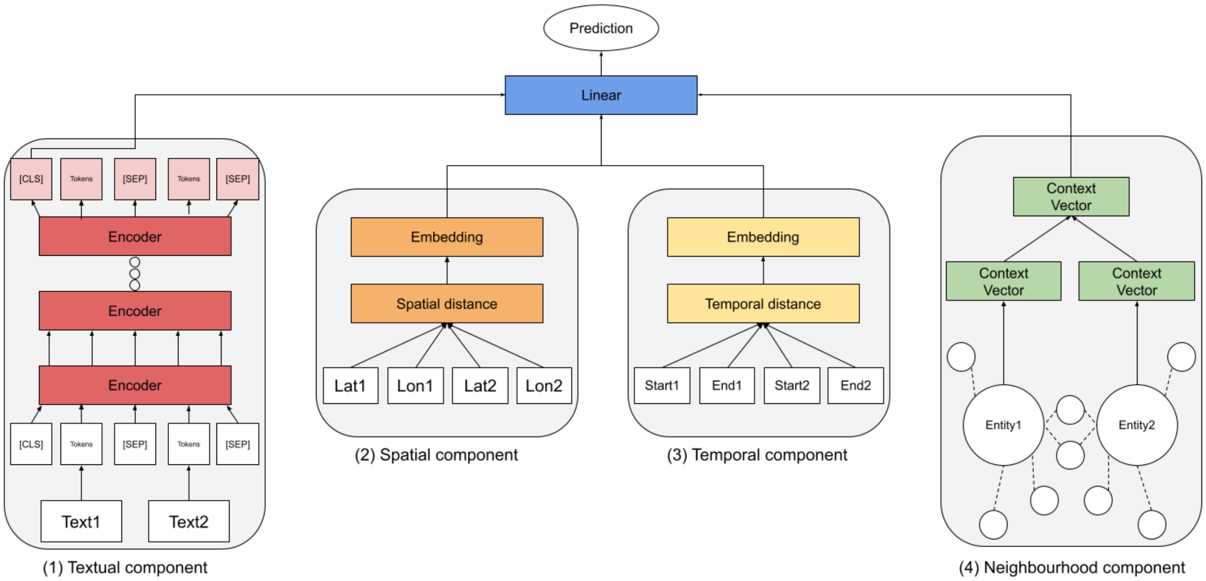

Figure 3. Overview of the model used for evaluation. (1) Is used for textual input i.e. toponym name, variants of the name and other textual information about the toponyms. (2) Is the spatial component that uses the coordinates of toponyms to calculate a distance between them. (3) Is the temporal component that uses information about when the toponyms are from to calculate the temporal difference between them. (4) Lastly there is the neighborhood component that embeds information about the two toponyms five closest neighbors.

5.4.1. How much textual information should be used?.

As shown earlier, each entity in the dataset has a number of textual attributes $\{name, variants, description\}$. In this experiment, we will see how much information should be used as input to get the best results when matching toponyms. For this experiment the full dataset was used together with the mBERT model, and the results can be seen in Figure 4.

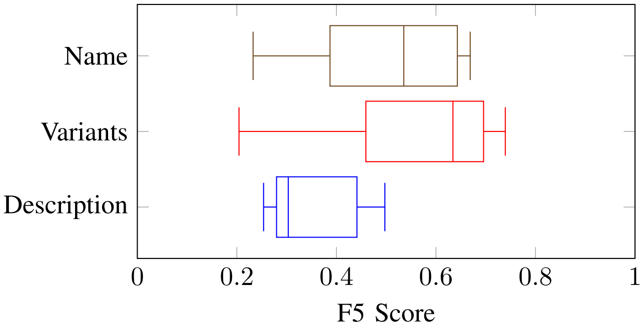

Figure 4. F5 scores from testing how many textual attributes should be used in the model. Variant also includes the Name attributes and Description includes both Variant and Name.

As can be seen from the results in Figure 4, the best number of textual attributes to use are *name* and *variants*. The results will be discussed further in Section 5.5. But for all the following evaluations only *name* and *variants* will be used as inputs for the textual component.

5.4.2. Comparing BERT models.

In this experiment asBERT, asemBERT and mBERT will be compared using the evaluation model described in Figure 3. Furthermore, the dataset is trimmed to only include toponyms that has a valid Spatial attributes, this is the case for about $\frac{5}{6}$ of the toponyms. The results can be seen in Figure 5.

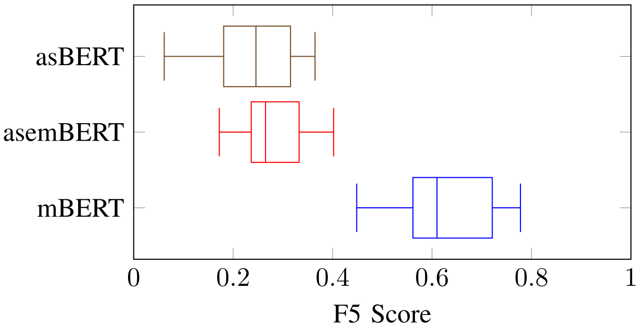

Figure 5. F5 scores from evaluating each BERT based model as the textual component on the task-specific dataset where only toponyms with valid coordinates are used.

Because of the surprising result in Figure 5, the `Recall` and `Precision` of the experiment will be shown in Figure 6 and Figure 7 respectively to further our understanding of the results.

5.4.3. Other pre-trained models.

Because of the quite surprising results of the previous experiment in Section 5.4.2, other pre-trained models will be tried against mBERT. Two versions of mBERT exist, one is cased and the other is uncased. Arbitrarily the cased one was chosen for this project even though Hebrew and Arabic does not use upper and lowercase letters like in English. The

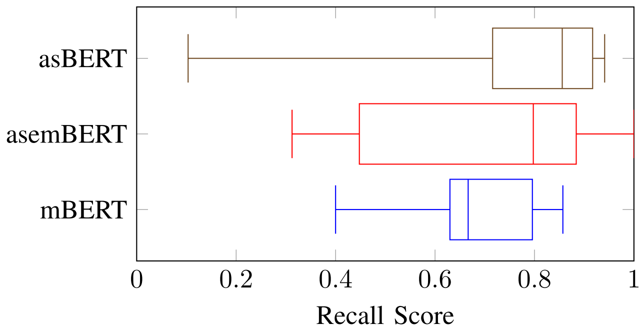

Figure 6. Recall scores from evaluating each BERT based model as the textual component on the task-specific dataset where only toponyms with valid coordinates are used.

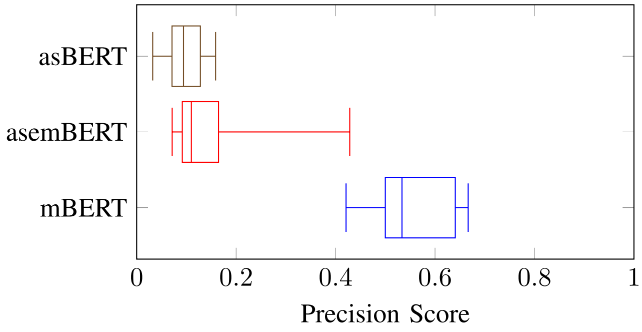

Figure 7. Precision scores from evaluating each BERT based model as the textual component on the task-specific dataset where only toponyms with valid coordinates are used.

third and final pre-trained model that will be used in this experiment is XLM-R [29] which is RoBERTa's equivalent of mBERT that has been shown to outperform mBERT. For this experiment the full dataset will be used, just like in the textual experiment in Section 5.4.1. The results from this experiment can be seen in Figure 8 showing the `F5` score, Figure 9 showing `Recall` scores and Figure 10 showing `Precision` scores.

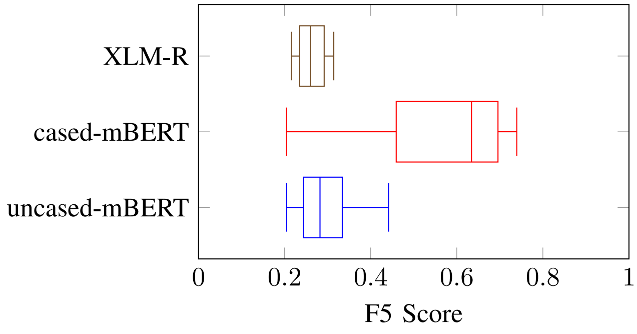

Figure 8. F5 scores from testing performance between three different pre-trained models using the full task-specific datasets.

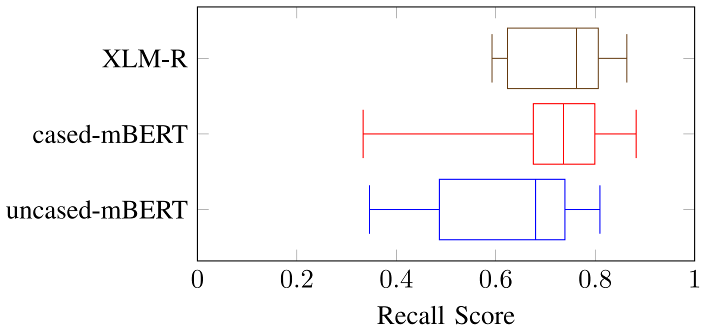

Figure 9. Recall scores from testing performance between three different pre-trained models using the full task-specific datasets.

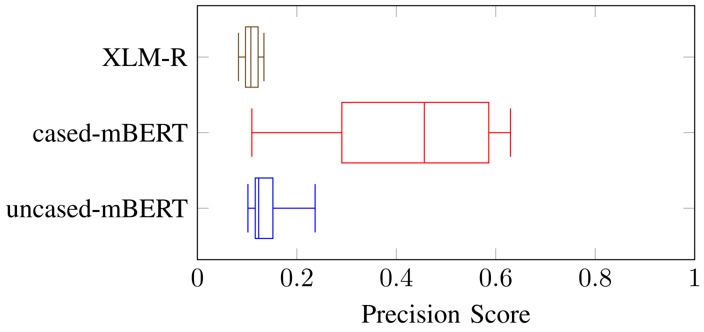

Figure 10. Precision scores from testing performance between three different pre-trained models using the full task-specific datasets.

5.4.4. Removing components.

For this experiment each of the components will be removed from the evaluation model shown in Figure 3 to see how it affects performance. For this experiment only toponyms with Spatial attributes were used together with the mBERT model, as it has been shown to be the best performing model. The reason that the dataset was not restricted to only toponyms with Spatial and Temporal attributes is because the majority of toponyms is missing at least one of the Temporal attributes. The results of this experiment can be seen in Figure 11.

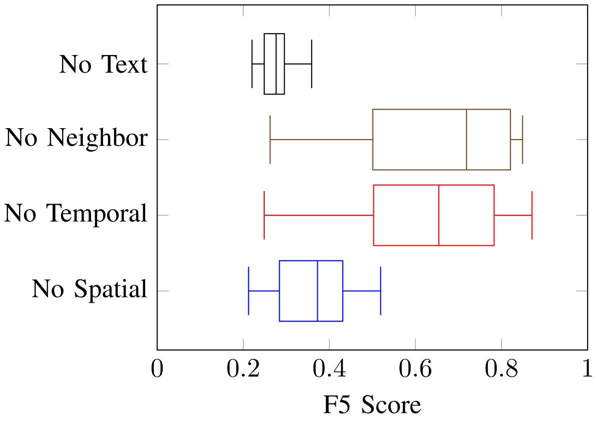

Figure 11. F5 scores from evaluating how performance is when removing components from the evaluation architecture using only toponyms with spatial information from the task-specific datasets.

5.5. Discussion of Results

The results presented in the previous section were quite surprising as it was quite confidently shown that the pre-trained mBERT model outperformed every other model. As was shown in Section 5.4.2 mBERT confidently outperformed both models created for this paper. This is quite surprising as mBERT has not been trained on a single ancient manuscript in either Arabic or Hebrew compared to both asBERT and asemBERT, who also have a lower perplexity score on the pre-training datasets conprising ancient Hebrew and Arabic. Furthermore, by looking at the `Recall` and `Precision` it can be seen that both asBERT and asemBERT have similar `Recall` scores compared to mBERT but much lower `Precision` scores. This shows that they are making a lot of guesses giving them a high `Recall` as they hit many matches, but on the other hand the low `Precision` show that they mostly predict incorrect matches From these results, it could seem like the textual input is not complex enough to warrant specialized models like asBERT and asemBERT or there is some other problem that can be hard to see because of the black-box architecture making it harder to analyze. Just as surprising was the results from Section 5.4.3 where mBERT was compared to other similar pre-trained models. Here the cased-mBERT very confidently outperformed both other pre-trained models giving results similar to the experiment using this papers proposed models. This results was surprising for a number of reasons. Arabic and Hebrew do not use upper and lowercase like English, so one's intuition might say that having an uncased model and tokenizer would leave room for more information as the case can be disregarded, but this was not what happened. The cased-mBERT also very confidently outperformed XLR-R, which on paper is the better model as it has been shown to perform better on a number of cross-lingual benchmarks. All other models besides cased-mBERT had very similar performance compared to each-other. Further testing and research are needed to fully understand these results and why mBERT is so much better. It would be beneficial to work more with the datasets used in the evaluation to find all the matches present in the datasets, as it could negatively impact the models learning abilities. Furthermore, an analysis looking at the matches that mBERT were able to find, that the other models could not identify could maybe help further our understanding of these results.

As for the evaluation model, it can be seen in Section 5.4.4 that the Textual and Spatial components are the most important, as F5 scores are reduced quite a bit when they are removed. Surprisingly, it looks like performance has increased as either the Temporal or Neighborhood component has been removed. The improved performance when removing Temporal is not that surprising because of the missing data in the datasets. But on the other hand removing the Neighborhood component improving performance is surprising, but this could indicate that the current implementation is lacking.

6. Conclusion

In this work we presented two new BERT based models called asBERT and asemBERT that were pre-trained on ancient manuscripts in Hebrew and Arabic. This was done with the intent of seeing the feasibility of using BERT based models to connect toponyms in different languages and scriptures. Surprisingly the evaluation showed that using a pre-trained model called mBERT from Google in its cased version resulted in far better results, even compared to other pre-trained models like XLM-R from Facebook, which on paper outperforms mBERT. We presented some theories to why this might be the case, but more research and work is needed to conclude why this was the case.

Acknowledgment

The author would like to thank Tomer Sagi for his supervision and help throughout this project. The MEHDIE project for the Hebrew dataset and inputs. OpenITI for the Arabic dataset. Sinai Rusinek and Moran Zaga for helping with looking over the false positives generated by mBERT, so that we could extend the task-specific datasets and improve our results.

References

[1] MEHDIE, "Mehdie project - about us," https://mehdie.org/, [Online; accessed 15-February-2024].

[2] K. Grossner and R. Mostern, "Linked places in world historical gazetteer," in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Geospatial Humanities*, ser. GeoHumanities '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 40–43. [Online]. Available: https://doi.org/10.1145/3486187.3490203

[3] P. Virga and S. Khudanpur, "Transliteration of proper names in cross-lingual information retrieval," in *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*. Sapporo, Japan: Association for Computational Linguistics, Jul. 2003, pp. 57–64. [Online]. Available: https://aclanthology.org/W03-1508

[4] T. Sagi, M. Zaga, S. Rusinek, M. R. Fekete, J. Bjerva, and K. Hose, "Utilizing phonetic similarity for cross-source and cross-language toponym matching - a benchmark and prototype," Mar. 2024. [Online]. Available: https://www.researchsquare.com/article/rs-4136375/v1

[5] B. Zhou, L. Zou, Y. Hu, Y. Qiang, and D. Goldberg, "Topobert: a plug and play toponym recognition module harnessing fine-tuned bert," *International Journal of Digital Earth*, vol. 16, no. 1, p. 3045–3064, Oct. 2023.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[7] M. V. Koroteev, "Bert: A review of applications in natural language processing and understanding," *ArXiv*, vol. abs/2103.11943, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232307525

[8] D. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 604–624, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:51872504

[9] A. Adhikari, A. Ram, R. Tang, W. L. Hamilton, and J. Lin, "Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, S. Gella, J. Welbl, M. Rei, F. Petroni, P. Lewis, E. Strubell, M. Seo, and H. Hajishirzi, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 72–77. [Online]. Available: https://aclanthology.org/2020.repl4nlp-1.10

[10] H. Zhang, J. Cai, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, M. Bansal and A. Villavicencio, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 789–797. [Online]. Available: https://aclanthology.org/K19-1074

[11] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4996–5001. [Online]. Available: https://aclanthology.org/P19-1493

[12] S. Wu and M. Dredze, "Are all languages created equal in multilingual BERT?" in *Proceedings of the 5th Workshop on Representation Learning for NLP*, S. Gella, J. Welbl, M. Rei, F. Petroni, P. Lewis, E. Strubell, M. Seo, and H. Hajishirzi, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 120–130. [Online]. Available: https://aclanthology.org/2020.repl4nlp-1.16

[13] J. Libovický, R. Rosa, and A. M. Fraser, "How language-neutral is multilingual bert?" *ArXiv*, vol. abs/1911.03310, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:207847232

[14] K. K, Z. Wang, S. Mayhew, and D. Roth, "Cross-lingual ability of multilingual bert: An empirical study," *ArXiv*, vol. abs/1912.07840, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:209183618

[15] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, , J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: Bert for finnish," *arXiv*, 2019. [Online]. Available: http://arxiv.org/abs/1912.07076

[16] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: a tasty French language model," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 7203–7219. [Online]. Available: https://aclanthology.org/2020.acl-main.645

[17] S. Rönnqvist, J. Kanerva, T. Salakoski, and F. Ginter, "Is multilingual BERT fluent in language generation?" in *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, J. Nivre, L. Derczynski, F. Ginter, B. Lindi, S. Oepen, A. Søgaard, and J. Tidemann, Eds. Turku, Finland: Linköping University Electronic Press, Sep. 2019, pp. 29–36. [Online]. Available: https://aclanthology.org/W19-6204

[18] M. Majadly and T. Sagi, "Dynamic ensembles in named entity recognition for historical arabic texts," *WANLP 2021*, p. 115–125, 2021.

[19] M. Zaga, T. Sagi, S. Rusinek, E. Lev, and M. Lavee, "A rule-based approach in geohistorical analysis: The case of medieval muslim," Oct. 2024.

[20] P. Singh, G. Rutten, and E. Lefever, "A pilot study for BERT language modelling and morphological analysis for ancient and medieval Greek," in *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, S. Degaetano-Ortlieb, A. Kazantseva, N. Reiter, and S. Szpakowicz, Eds. Punta Cana, Dominican Republic (online): Association for Computational Linguistics, Nov. 2021, pp. 128–137. [Online]. Available: https://aclanthology.org/2021.latechclfl-1.15

[21] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, "Greek-bert: The greeks visiting sesame street," in *11th Hellenic Conference on Artificial Intelligence*, ser. SETN 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 110–117. [Online]. Available: https://doi.org/10.1145/3411408.3411440

[22] Y. Chang, P. Zhu, C. Wang, and C. Wang, "Automatic word segmentation and part-of-speech tagging of ancient chinese based on bert model," in *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, R. Sprugnoli and M. Passarotti, Eds. Marseille, France: European Language Resources Association, Jun. 2022, p. 141–145. [Online]. Available: https://aclanthology.org/2022.lt4hala-1.20

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:198953378

[24] M. Davari, L. Kosseim, and T. Bui, "Timbert: Toponym identifier for the medical domain based on bert," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, p. 662–668. [Online]. Available: https://aclanthology.org/2020.coling-main.58

[25] P. Balsebre, D. Yao, G. Cong, and Z. Hai, "Geospatial entity resolution," *Proceedings of the ACM Web Conference 2022*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248367485

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[27] M. Joshi, O. Levy, L. Zettlemoyer, and D. Weld, "BERT for coreference resolution: Baselines and analysis," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5803–5808. [Online]. Available: https://aclanthology.org/D19-1588

[28] P. Balsebre, D. Yao, G. Cong, and Z. Hai, "Geospatial entity resolution," in *Proceedings of the ACM Web Conference 2022*. Virtual Event, Lyon France: ACM, Apr. 2022, p. 3061–3070. [Online]. Available: https://dl.acm.org/doi/10.1145/3485447.3512026

[29] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747