Universiteit Utrecht

Applied Data Science
Natural Sciences
Utrecht University
3508 TC Utrecht, The Netherlands

## Subject

Knowledge graph expansion using cloze statements by leveraging language models

Author: Ignasi Oliveres Torrescassana

July 19, 2022

| | |
|---|---|
| Author: | **Ignasi Oliveres Torrescassana** |
| St. Number: | 2834928 |
| Email: | i.oliverestorrescassana@students.uu.nl |
| Supervisors: | Dr. M.W. (Mel) Chekol, Utrecht University |
| | D.S. (Duygu) Islakoglu, Utrecht University |
| | Dr. A.A.A. (Hakim) Qahtan, Utrecht University |

# Abstract

Knowledge Graphs are useful for representing information by using a collection of facts. If temporal information is added to a Knowledge Graph, it is expanded into a Temporal Knowledge Graph. Pre-trained Language Models are trained with large datasets and could act as a Knowledge Graph if they learn enough in the pre-training stage. Although studies have been done in the past to understand if pre-trained Language Models can act as Knowledge Graphs, to the best of our knowledge, there is still no study about whether they can be used for Temporal Knowledge Graph expansion. This thesis uses cloze statements to understand if state-of-the-art pre-trained Language Models can be used for expanding a Knowledge Graph into a Temporal Knowledge Graph. In order to do this, several templates have been created to transform facts in a Knowledge Graph into cloze statements to study the performance, robustness and ability to reason of 5 of the most important pre-trained Language Models. The results revealed that, at the moment, pre-trained Language Models are not reliable enough to be used for expanding a Knowledge Graph into a Temporal Knowledge Graph.

**Key Words:** Knowledge Graph; Temporal Knowledge Graph; Natural Language Processing; Language Models; Template; Cloze statements.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Natural Language Processing (NLP) is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis to achieve human-like language processing for a range of tasks or applications, as explained by Liddy (2001). In other words, NLP combines linguistics and machine learning (ML) to give the ability to understand human language to a computer program. At the moment, there exist several language models (LMs) that can do different tasks. Some examples of the most common NLP tasks are sentence classification (Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown (2019)), generating text content (Iqbal and Qureshi (2020)), extracting an answer from a text (Abacha and Zweigenbaum (2015)), and translating text from one language to another (Zhang and Zong (2015)).

A way to represent the knowledge behind language is with a knowledge graph (KG). KGs consist of facts that represent some information. Facts are composed of entities, relationships, and semantic descriptions, and KGs can be represented as networks containing all these items. They are a structured representation of facts. Entities (e.g. Albert Einstein) can be both real-world objects and abstract concepts, and the relation between them is represented by the relationships of the fact (i.e. edges of the network). Both entities and relationships have types and properties with a defined meaning represented by the semantic descriptions. All this information that creates a fact can be seen as a triple in the form of (head, relation, tail), i.e. (h, r, t) or (subject, predicate, object). An example of a factual triple that expresses knowledge is (Albert Einstein, WinnerOf, Nobel Prize). The result of a graph representation of knowledge is a directed network where the nodes are the entities, and the edges are the relationships between them (Ji, Pan, Cambria, Marttinen, and Yu (2022)).

Ji, Pan, Cambria, Marttinen, and Yu (2022) also explain that triples representing a fact can be extended to a quadruple by adding temporal information. In this case, the triplet (h, r, t) is converted to a quadruple in the form of (h, r, t, T), where T represents the temporal information of when the fact happened. The temporal information can be a single time (for instance, a year) but also a period of time. In this case, the temporal information of the quadruple contains the period's start and end. Weikum et al. (2020) specify that, since the real world evolves and

changes over time, its information also changes. Therefore, temporal knowledge graphs (TKGs) must include and consider this evolution. Adding this temporal information to a KG is the KG expansion to a TKG.

There are several studies about whether the existing pre-trained LMs can act as a KG, but there is no agreement about the answer. For instance, in Petroni, Rocktäschel, Lewis, Bakhtin, Wu, Miller, and Riedel (2019) research, they found that there exist some pre-trained models, such as BERT (Devlin et al. (2018)), that can act as a KG. As they state, since the pretraining process of these models is done with a large amount of data, they can learn (i.e. store) the knowledge present in this data. They also explain that LMs have many advantages over the KGs, such as no need for human supervision to train, they don't need any schema engineering, or they are easy to expand. This last advantage can be done by querying "fillin-the-blank" cloze statements (Taylor (1953)), using Masked Language Models (MLMs). These models receive a sentence with blank (i.e. masked) words which they have to predict. For example, they receive the sentence *"Paris is the capital of [MASK]."* and they have the predict the masked word *[MASK]*, in this case, France.

This thesis tries to give more insights into the uncertainty about whether pre-trained LMs can act as KG. More specifically, the main goal of this work is to check whether pre-trained LMs are robust to expand KGs to TKGs by using cloze statements.

To be able to do that, this work aims to combine KGs and NLP models to check the State-of-the-art methods' performance with temporal facts. More specifically, three research questions are proposed:

1. What is the performance of the existing NLP pre-trained models on temporal cloze statements?
2. Are NLP pre-trained models able to reason about time?
3. Are NLP pre-trained models robust?

The TKG containing Wikipedia information (Wikidata12k) is used to answer the questions defined above.

# Chapter 2

# Related Work

Some research about the topic regarding this thesis has been already done. In this section, an overview of it is described.

Whether pre-trained LMs can act as a KG has been studied in recent years. However, there is not a complete agreement on the answer. Petroni, Rocktäschel, Lewis, Bakhtin, Wu, Miller, and Riedel (2019) report in their study that the BERT model can recall factual knowledge with no need of fine-tuning, but just by using a prompt to retrieve it, such as *"Einstein was born in ____ "* to query the place of birth of Einstein. While Petroni, Rocktäschel, Lewis, Bakhtin, Wu, Miller, and Riedel (2019) created the prompts manually, Jiang, Xu, Araki, and Neubig (2019) created a method to automatically generate different prompts. The fact that the BERT model can contain factual knowledge means that no changes in the model need to be made to achieve a good performance. Therefore, the model's parameters already contain enough knowledge, being an unsupervised process. However, Cao, Lin, Han, Sun, Yan, Liao, Xue, and Xu (2021) question the origin of the decent performance in Petroni, Rocktäschel, Lewis, Bakhtin, Wu, Miller, and Riedel (2019) results. They studied whether MLMs could be reliable knowledge graphs. They found out that the predictions were prompt-biased, meaning that they correlate with the prompt and not with the subject (i.e. entity). Despite these two studies, among others, are related to the topic of this thesis, none of them studies if pre-trained LMs can be used for KG expansion to TKG.

Saxena, Chakrabarti, and Talukdar (2021) also checked the performance of pre-trained LMs on TKGs. They presented the largest Temporal KGQA dataset at that moment and proposed a transformer-based solution that increased by 120% the accuracy of the best method previously existing. However, they used question answering for their study and not cloze statements, which is the method used in this thesis.

Finally, Qin, Gupta, Upadhyay, He, Choi, and Faruqui (2021) studied whether pre-trained LMs could reason. They found out that even the models with the best performance were far from human performance. Their qualitative error analyses could also conclude that the mod-

els fail to reason over the context. Despite Qin, Gupta, Upadhyay, He, Choi, and Faruqui (2021) investigating a similar topic as the one I am investigating in this thesis, they checked the ability of the pre-trained LMs to reason with temporal information in dialogs by checking the context of the dialog itself, but not about time intervals on cloze statements, as I do in this work.

Although there exist several studies related to the topic addressed in this work, none of them contributes in the same way as this one, which is checking how reliable are pre-trained LMs to be used for a KG expansion to a TKG.

# Chapter 3

# Data Preparation

## 3.1 Dataset

The primary dataset used for the analysis is the train set of Wikidata12k (Jain, Rathi, Mausam, and Chakrabarti (2020)). This dataset contains information from Wikipedia in a KG format. Specifically, it is a TKG since it has a time interval associated with each triple. It contains 12544 unique entities and 24 relations id creating a total of 32497 different observations. These observations can be differentiated into four types of events depending on the time period. Some observations are considered a time event or time instant (i.e. Year Event). Those are the observations that have the same start and end date. There are 14099 quadruples in this subset. In the second and third types, there are observations with only the start date or the period's end date, considered an open time interval. It can be a right open or left open interval, depending on which year (start or end) is null. The left open interval subset (start year is null) contains 1273 quadruples, and the right open interval one (end year is null) is 4089. The fourth type of quadruples is the one that consists of a closed time interval, having different (and not null) start and end dates. In this last subset, there are 13036 quadruples. This information can be found in the table 3.1.

| **Subset** | **Amount of quadruples** |
|---|---|
| Closed Interval | 13036 |
| Left Open Interval | 1273 |
| Right Open Interval | 4089 |
| Year Event | 14099 |

Table 3.1: Subsets of Wikidata12k.

## 3.2 Data Cleaning

Dates attributes in the Wikidata12k train dataset are given in the format *YYYY-##-##*. Hence, a conversion of both start and end date is done to a *YYYY* format, since to answer the first research question proposed (checking pre-trained LMs performance on temporal cloze statements), these years need to be masked to be able to predict them afterwards.

Another step that needs to be done before running the models is deleting the quadruples with anomalies. These can be quadruples with the end year lower than the start year or those without entities' descriptions. As explained in later sections of this thesis, quadruples can be expressed in natural language format. Templates are used to do this process. Templates are sentences in a natural language format that allows expressing the information of a single quadruple in different ways but still having the same meaning. They can also be called prompts. Some of these templates are created by adding context to them. This is done by adding the description of the head and tail of the quadruple in the sentence. However, some entities do not have a description in the Wikidata database. Since one of the analyses of this study consists in checking if there are differences between the templates proposed, the quadruples containing entities without description are deleted to avoid having biased results.

## 3.3 Dataset Split

The performance of the different models and templates is checked among the four data types in Wikidata12k (Closed Interval, Left Open Interval, Right Open Interval, Year Event). After cleaning the data, the size of each of these subsets can be seen in the table 3.2.

| **Subset** | **Amount of quadruples** | **Amount of templates** |
|---|---|---|
| Closed Interval | 12276 | 36 (8) |
| Left Open Interval | 1214 | 16 |
| Right Open Interval | 3681 | 16 |
| Year Event | 11728 | 8 |

Table 3.2: Subsets of Wikidata12k after preprocessing. In the Closed Interval subset, the value inside the parenthesis indicates the amount of duration templates.

# Chapter 4

# Methods

## 4.1 Models

A Transformer is a ML model used for NLP tasks presented by Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin (2017). It uses the self-attention mechanism, which allows the model inputs to interact with each other and discover which input the model should pay more attention to. This type of model has an encoder-decoder architecture. In this architecture, the first stage is the encoder. It receives the input of the model (i.e. a sequence of words) and outputs a numerical representation for each word. In this stage, all the words in the sequence affect every word using the previously mentioned self-attention mechanism. The encoder outputs are sent to the second stage of the process, the decoder. The decoder receives the words' numeric representation from the encoder but also the start of the sequence word. The difference in this stage is that the decoder uses a masked self-attention mechanism. Therefore, not all words are affected by every word in the initial sentence, but only by the ones before it. That is, the decoder uses the output from the encoder and the start of the sequence word to predict the first word. Afterwards, it does the same process to predict the second word but also uses the first word predicted as an input.

All the pre-trained LMs used in this work are variants of the BERT model, introduced by Devlin, Chang, Lee, and Toutanova (2018). BERT stands for Bidirectional Encoder Representations from Transformers. The BERT model is transformer-based, and it is pre-trained in a self-supervised way, meaning with no human actions taken for labelling the texts but with an automatic process to generate inputs and labels for the texts. Moreover, the BERT model is specifically pre-trained with the MLM objective. It randomly masks 15% of the input tokens and then tries to predict the token id based on the context. As the name indicates, it is a bidirectional encoder since it uses the token's left and the right context to predict it, allowing pre-train a deep bidirectional Transformer. This method differs from the left-to-right LM pre-training, which only uses the previous context of the word (left part of the sentence) to predict it. The BERT model is also pre-trained with the next sentence prediction (NSP) objective.

Five different models are used in this thesis. The first of them is the BERT base model cased. This model is case-sensitive. For example, it can understand that *english* and *English* are different words. That is the main difference between the second model used, the BERT base model uncased, which does not differentiate between uppercase or lowercase letters and lowercase all the words before tokenizing them. The third model considered for the analysis is the DistilBERT model uncased (Sanh, Debut, Chaumond, and Wolf (2019)). This model is a distilled version of the BERT model and is smaller, cheaper, faster and lighter. Specifically, it reduces by 40% the size of the BERT model, and it is 60% faster, but it still achieves 97% of BERT performance. The DistilBERT model uses the compression technique of Knowledge distillation, which, as defined by Bucila, Caruana, and Niculescu-Mizil (2006), allows a student and more compact model (DistilBERT) to reproduce the behaviour of a larger model, that is, the teacher (BERT). Finally, this model is also uncased, so it is not case-sensitive. Another model used is the RoBERTa base model (Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer, and Stoyanov (2019)). The name of this model comes from Robustly optimized BERT approach. As it indicates, it is a more robust and optimized approach of the BERT model by modifying the pre-training process and leading to better end-task performance. All these first four models mentioned are trained in two large datasets. One is the dataset containing all the information from Wikipedia, and the second is the *Book Corpus*, which contains texts from books. The last model used is the XLM-RoBERTa model (Conneau, Khandelwal, Goyal, Chaudhary, Wenzek, Guzmán, Grave, Ott, Zettlemoyer, and Stoyanov (2019)). It is a multilingual version of the RoBERTa model, and it is pre-trained with data containing 100 languages. The dataset used to pre-train the XLM-RoBERTa model is Common Crawl, which gathers information from the Internet.

## 4.2 Knowledge Graph to Textual Data

The NLP models described need text in natural language format to feed them to retrieve a prediction for the masked word. Therefore, a transformation process must be done to convert the TKG information into natural language sentences. Since the Wikidata12k dataset is given with a number for entities and relations instead of the original id, two files that convert these numbers to the ids are needed for the entities and relations, respectively. The next step to building sentences in natural language format is to have the label and description of each entity and relation. This is done with SPARQL. This tool can retrieve both the label and the description of each entity and relation from the Wikidata database. These labels and descriptions help create different types of templates with the same quadruple information. Multiple templates are also helpful for further analyses, such as understanding if NLP models work better when they have more context or if they understand better a specific way of creating a sentence, among others.

## 4.3 Templates Creation

The quadruple (h, r, t, T) of a TKG contains the head, relation, tail and temporal information of a fact. These items can be modified to create different types of templates (i.e. prompts) using the same quadruple. With entities (head and tail), the label or both the label and the description can be used in the construction of natural language sentences in order to give more context to it. With the relation, the raw label of the relation can be used, but also a more human version. That is, using the correct verbal tense in each situation, adding the correct articles etc. Finally, for the temporal part of the quadruple, different versions can be created with the same meaning. An example of these different versions can be "since [MASK] until [MASK]", "from the year [MASK] to the year [MASK]", or "between the years [MASK] and [MASK]" for the Closed Interval subset. Moreover, duration templates are created for the Closed Interval subset to answer one of the main research questions. This is done to understand if NLP models can reason. In this case, the models will have to predict the number of years inside the interval (masked) instead of its start and end years. Again, different templates can be created expressing the same information. When retrieving the masked years, the models give them as an output with a numeric format (e.g. *1925*). However, for a simple number as the interval duration, the models can retrieve the prediction either in a numeric format (e.g. 8) or in a text format (e.g. *eight*). For this situation, both formats are accepted, and a predicted value is considered correct if it is in number or text format. This procedure of searching the prompt with better outcomes is called prompt engineering (Reynolds and McDonell (2021)).

Combining each element's versions in a quadruple gives several templates for every fact in each of the subsets of Wikidata12k. Specifically, for the Closed Interval, there are 36 different possible templates for predicting the start and end years of the interval, as well as eight more for predicting the duration of each interval. In the case of the Left Open Interval and Right Open Interval subsets, both of them have 16 possible templates. Finally, the Year Event one contains 8 templates, as shown in table 3.2.

## 4.4 Templates Selection

Additionally to all the aforementioned templates that can be applied to every quadruple in each subset, five models are run on all of them. This leads to very high computational costs and time. Due to the lack of computational power and resources, a reduction in the number of templates is needed. To do so, six templates are selected (for the Closed Interval, also two templates for the duration) for each of the four subsets. To do this filtering process without including any bias, a set of 100 observations is randomly selected for each subset. For these sets, all templates created are run for the five models proposed, and the six best performing in each set are selected. Once the filtering process is done, the five models can be rerun with all the observations in each subset. In this case, for the six final templates. Table 4.1 shows the structure of the selected templates for all the subsets.

| Subset | Templates |
|---|---|
| Closed Interval | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “from the year” [MASK] “to the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “from the year” [MASK] “to the year” |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “between” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “between” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “between the years” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “from” [MASK] “to” [MASK] |
| Closed Interval Duration | [HEAD][RAW_RELATION][TAIL] ”for” [MASK] ”years” |
| | [HEAD][TEMPLATE_RELATION][TAIL] ”for” [MASK] ”years” |
| Open Interval | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] ”since”/“until” [MASK] |
| | [HEAD] [RAW_RELATION][TAIL] ”since”/“until” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “starting from the year”/“finishing in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “starting from the year”/“finishing in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “since the year”/“until the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] ”since”/“until” [MASK] |
| Year Event | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “in” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “in” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “in the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “in the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “in” [MASK] |

Table 4.1: Best 6 templates for each subset. For the Open Interval subset, the first version of the temporal part of the sentence (e.g. ”since”) is for the templates of the Right Open Interval. The Left Open Interval has the same templates using the second version of the temporal part of the sentence (e.g. ”until”).

## 4.5 Performance and Robustness Measures

### 4.5.1 Performance

The accuracy measure is used to check how well each pre-trained LM works on predicting temporal cloze statements. In this case, accuracy is calculated as the percentage of correct predictions. The formula of the accuracy measure is given in equation 4.1.

$$Accuracy = \frac{TotalCorrectPredictions}{TotalNumberOfPredictions} \cdot 100 \tag{4.1}$$

All the models used can retrieve not just one predicted value but many. These prediction values are given sorted by a score that the model gives to each prediction to consider less or more probable that prediction to be correct. Hence, more than one predicted value is used to calculate several accuracy measures for each prediction. Four accuracy measures are used: top1, top3, top5 and top10. For each of them, the prediction for an observation is considered a correct prediction (equation 4.2) if the observed value is in the first $k$ predictions values depending on the measure used.

$$CP_k = \sum_{i=1}^{\#Obs} Val_{i,k} \tag{4.2}$$

where $Val_{i,k}$ is a binary value being true if the observed value is in the top $k$ prediction values of the prediction. The 4 aforementioned accuracy measures are calculated with the formula shown in the equation 4.3, with k being 1, 3, 5 and 10.

$$Accuracy = \frac{CP_k}{TotalNumberOfPredictions} \cdot 100 \tag{4.3}$$

For the Left Open, Right Open intervals and Year Event subsets, the process is to mask the year in the sentence and retrieve if the prediction is correct or not. However, with the

quadruples in the Closed Interval, the masking and evaluation processes can be done in multiple ways. First of all, the start and end year of the interval can be masked together or separately. Masking them together means that the model has as an input one natural language sentence with the start year and the end year of the interval masked, and it retrieves two predictions simultaneously, one for each masked year. The other option is to mask them separately. In this case, the model has two natural language sentences as inputs. First, it receives a sentence with the start year of the interval masked and predicts it. After it, it does the same with a sentence with the end year masked. When masking separately, the LM has more context and, therefore, more information for the masked word since the year that is not predicted remains unmasked and known for the model. For the evaluation process, there are several ways to assess the performance of a model. The different prediction accuracy measures can be calculated for only the start year of the interval, the end year, or the start and end year at the same time. For the third option, a prediction is only considered correct when both start and end year predictions are correct.

### 4.5.2 Robustness

To check a model's robustness, the randomness of its predictions is calculated. To do so, the standard deviation measure is used. For every observation prediction, the standard deviation of the ten retrieved guesses for the prediction is calculated. This is represented by equation 4.4. Note that in some cases, the prediction retrieved by the model can be a regular word instead of a year/number. For the standard deviation calculation, only the years or numbers are used. The next step is to average the different standard deviations from the predictions of each model (equation 4.5).

$$Std_i = \sqrt{\frac{\sum (X - \mu)^2}{N}} \tag{4.4}$$

Where $Std_i$ is the standard deviation for one prediction, $X$ are the ten different predicted values retrieved for the prediction, $\mu$ is the mean of these ten values, and N is 10.

$$\overline{Std} = \frac{\sum Std_i}{TotalNumberOfPredictions} \tag{4.5}$$

## 4.6 Framework

The process described in this chapter can be seen in the figure 4.1. It shows how the source of information is a TKG and how, from this information, several templates are created after converting the available information to a natural language format. After masking the years in these templates, they are filtered, and these filtered ones are passed by the pre-trained LMs proposed. After, the performance is evaluated.

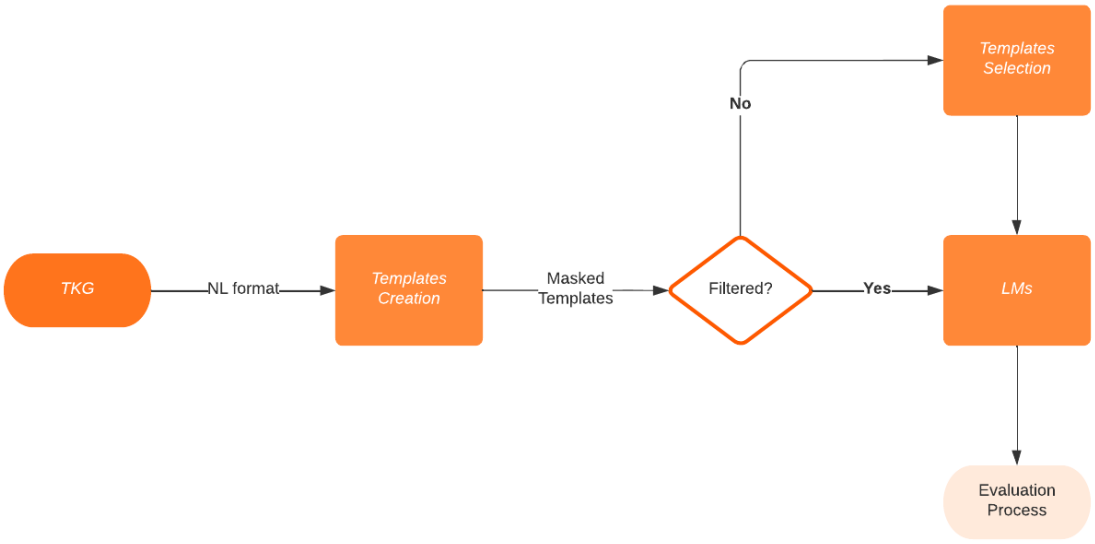


**Figure. 4.1:** Process Framework.

# Chapter 5

# Empirical Evaluation

## 5.1 Computational Cost

As aforementioned, the best six templates (2 for the duration templates) were filtered out of the total amount of templates for each subset shown in table 3.2 to deal with computational costs issues. After this selection, the run times for each model and subset are shown in the table 5.1. The Open Interval subset includes both the left and right open intervals. They were run together and split afterwards for the analysis.

| | Year Event | Open Interval | Closed Interval Masked Together | Closed Interval Masked Start Year | Closed Interval Masked End Year | Closed Interval Duration |
|---|---|---|---|---|---|---|
| **DistilBERT base uncased** | 56m 3s | 45m 18s | 19m 20s | 54m 18s | 55m 3s | 13m 52s |
| **BERT base cased** | 1h 30m 32s | 1h 13m 7s | 31m 11s | 1h 29m 11s | 1h 29m 52s | 20m 45s |
| **BERT base uncased** | 1h 30m 4s | 1h 15m 43s | 31m | 1h 28m 54s | 1h 29m 31s | 20m 38s |
| **RoBERTa** | 1h 49m 9s | 1h 38m 1s | 38m 40s | 1h 45m 38s | 1h 48m 17s | 25m 59s |
| **XLM-RoBERTa** | 4h 25m 58s | 3h 54m 51s | 1h 53m 2s | 4h 21m 3s | 4h 23m 22s | 1h 8m 32s |
| **Amount of observations** | 11728 | 4895 | 12276 | 12276 | 12276 | 12276 |

Table 5.1: Run time to predict for each model and subset.

For each subset, every model predicted the masked years for six templates (2 for the duration templates), resulting in 30 different predictions for every observation in the subset. As it can be seen in the table 5.1, the model that took longer is the XLM-RoBERTa.

## 5.2 Closed Interval Results

For the closed interval subset, the analysis is divided into two parts. First, the results of predicting the years of the interval are studied. After, the results predicting the duration of the interval are analyzed too.

### 5.2.1 Years Prediction

When comparing the two masking methods (together or separate), from the results shown in the table 5.2, it can be seen that masking the interval's start year and the end year separately gives better results in all the predictions than masking them together. This can be because

the models have more information when predicting with the years masked separately since they have the end year as a context when predicting the start year and the other way around.

| | Start&End Year | Start Year | End Year |
|---|---|---|---|
| **Masked Together** | 11.26% | 23.43% | 34.92% |
| **Masked Separate** | 56.93% | 65.88% | 76.04% |

Table 5.2: Average accuracy top10 of predicting both start and end year correctly, only the start year correctly, and only the end year correctly. The results are shown for the prediction done masking both start and end year of the interval together or separately.

The average performance of all the models and templates proposed reaches 76.04% accuracy top10 for the prediction of the end year of the interval when masking separately. With both masking methods, the highest accuracy values are when predicting the end year of the interval, while it decreases for predicting the start year of it. The lowest values are for predicting both the start and end year of the interval since a correct prediction means that both years of the interval are predicted correctly. To simplify all the analyses and explanations, from now on, when referring to the Closed Interval subset results, I am going to refer to the results of predicting both the start and end year of the interval masked separately.

### 5.2.2 Duration Prediction

In the Closed Interval subset, the models were also run to predict the duration of each interval, with the two templates filtered from all the duration templates proposed. The average accuracy top10, in this case, is also one of the highest, specifically, 54.17%.

### 5.2.3 Comparison

The interval duration's predictions using the duration templates and the predictions of the intervals' start and end years are the ones with the best performance. In the table 5.3, all the accuracy measures average values can be seen. These results show that the behaviour for both predicting methods is very similar, being the duration templates better for the accuracy top1 and top3. At the same time, the predictions of the start and the end year of the interval have better performance for the accuracy top10. For the accuracy top1, the duration templates have the best performance out of all the subsets studied, with a value of 5.86%.

| | Accuracy top1 | Accuracy top3 | Accuracy top5 | Accuracy top10 |
|---|---|---|---|---|
| **Pred. Start&End Year** | 3.30% | 14.47% | 29.49% | 56.93% |
| **Pred. Duration Templates** | 5.86% | 18.13% | 29.80% | 54.17% |

Table 5.3: Average accuracies when predicting Start&End year masked separately and when predicting the interval duration with the duration templates.

Besides the overall average results, an analysis comparing which observations groups have the best performance per each predicting method has been done to understand if both methods predicted the same type of observations well or not. The results are shown in the figure 5.1. The average accuracy per interval duration has been studied. Additionally, the performance for the decade when each fact happens has been analyzed too. 10161 observations have an interval

length of 10 years or lower, representing 82.7% out of the total amount of quadruples in the subset. From these 10161 observations, 95% are from the 1900s on. It can be seen that the shortest intervals (2 years, e.g. *[1995-1996]*) and the 2000s decade have the greatest amount of data.

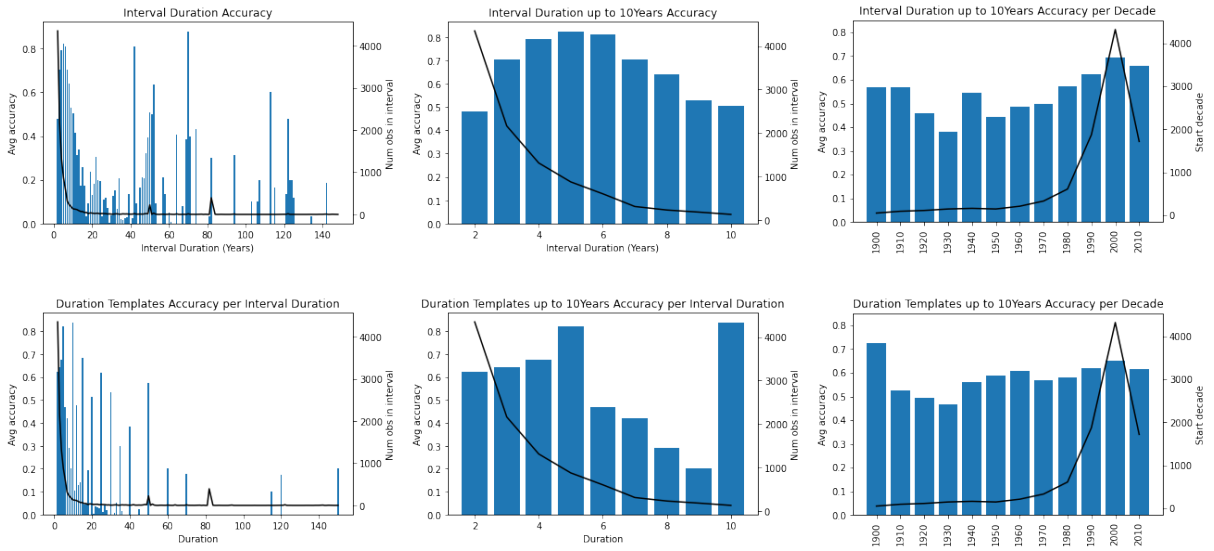


**Figure. 5.1:** The first row of figures shows the average top10 accuracy per the interval duration, per interval duration up to 10 years, and per decade of those intervals up to 10 years when predicting Start&End years masked separately. The second row shows the same information when predicting the duration of the interval with the duration templates. The black line indicates the amount of observations in each bar.

As figure 5.1 shows, there are some similarities but also several differences in which kind of quadruples have better results between interval duration predictions and the predictions of the start and the end year of each interval. In the graphs of the first column of the figure, it can be seen that in both cases, the observations with shorter intervals have the best performance. It decreases until the interval reaches 40 years, where it increases again. However, when predicting the start and the end year of the intervals, the accuracy decreases faster than the interval duration predictions. Also, after reaching the interval duration of 40 years, the accuracy increases again in several interval lengths when predicting the start and end year. In the interval duration predictions, there is only a peak in the 50 years length. Additionally, the graphs show that both methods have high accuracy values for the five years intervals when looking at the shorter intervals (second column in figure 5.1). However, the shape of the start and the end years of the intervals predictions accuracy top10 increases until the five-year interval and then decreases. At the same time, in the interval duration predictions, the performance for the ten-year length interval is also one of the highest.

On the other hand, the behaviour for both methods is also similar when checking which decade has better performance. In both cases, the average accuracy decreases until the 1930s and increases again after that decade. However, the increased slope of the interval duration predictions is flatter than when predicting the limiting years of the interval. Another difference is that the accuracy for the 1900s decade is the highest when predicting the interval duration.

## 5.3 Open Interval Results

The Open Interval subsets have, in general, a worse performance than the Closed Interval one. In this case, they reach an accuracy top10 value of 22%, as table 5.4 shows.

| | Accuracy top1 | Accuracy top3 | Accuracy top5 | Accuracy top10 |
|---|---|---|---|---|
| **Left Open Interval** | 4.31% | 10.33% | 14.82% | 22.59% |
| **Right Open Interval** | 2.39% | 7.57% | 11.73% | 22.06% |

Table 5.4: Average accuracies for right and left open interval subsets.

In the Open Interval subsets, there exist some differences between the Left Open and the Right Open Interval subsets. The Left Open interval subset is slightly better than the Right Open in all four accuracy measures. Moreover, regarding the average top1 accuracy, the Left Open Interval is the subset with the second-best results, behind the duration interval predictions.

In the figure 5.2, the distribution of the performance and the amount of data per decade can be seen for both the Left and Right Open Interval subsets. In terms of number of observations per decade, the shape is similar to the other subsets, increasing significantly in the last decades. However, in the Right Open Interval, there are two peaks in the 1920s and the 1960s. The majority of the quadruples in these decades have the relation of *"located in the administrative territorial entity"*. The leading cause of these two peaks is the creation of new departments in France during these two decades (3 and four departments respectively). These different departments were also divided into multiple communes or municipalities inside them. Therefore, several quadruples of the dataset, which are communes of these French departments, have their start date in the 1920s and 1960s.

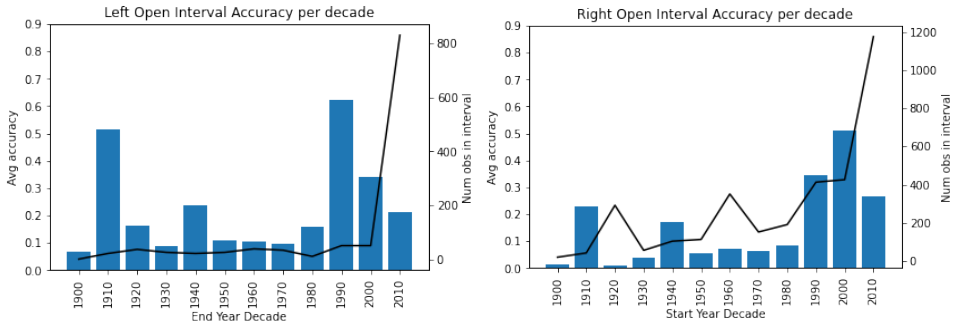


**Figure. 5.2:** Average accuracy top10 represented by the bars and number of observations represented by the black line per decade for Left and Right Open Interval subsets.

On the other hand, in terms of average top10 accuracy per decade, the shape for both subsets is similar, but with higher values for the Left Open Interval one. Also, in this subset, the performance for the 1910s is much better than the Right Open Interval one. While in the Right Open Interval subset, the decade with the best accuracy is the 2000s, in the Left Open one, it is the 1990s.

## 5.4 Year Event Results

The accuracy values of the Year Event subset are very similar to the Right Open Interval subset. In terms of accuracy, it is one of the worst out of all the subsets studied, as seen in table 5.5.

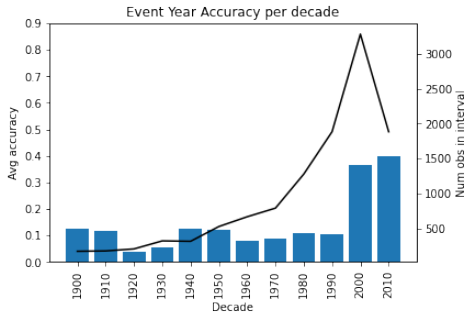


**Figure. 5.3:** Average accuracy top10 (bars) and number of observations per decade (black line) for Year Event subset.

| | |
|---|---|
| **Accuracy top1** | 2.14% |
| **Accuracy top3** | 6.89% |
| **Accuracy top5** | 11.30% |
| **Accuracy top10** | 22.12% |

Table 5.5: Average accuracies for Year Event subset.

The decades with the best performances are the 2000s and 2010s, but with accuracy values lower than the Closed Interval subset. Again, the highest peak of number of observations is close to the present, specifically in the 2000s.

## 5.5 Models

Until now, the overall results for each subset studied have been presented. However, a more detailed analysis has been done to understand which are the best models and if there exist differences between them among all the subsets. In this section, this deeper analysis is presented. This analysis is divided into two parts, one studying the accuracy performance of the models and another one analyzing their robustness.

### 5.5.1 Accuracy Performance

Regarding the accuracy performance, there are differences between the different subsets in general terms, as aforementioned, and between the models. These results can be seen in the table 5.6. Differences exist in the accuracy values within the subsets among the five models used and within each model among all the subsets.

| | Closed Interval | Closed Interval Duration | Right Open Interval | Left Open Interval | Year Event |
|---|---|---|---|---|---|
| **DistilBERT base uncased** | 67.81% | 54.95% | 29.80% | 28.76% | 26.26% |
| **BERT base cased** | 68.78% | 51.12% | 19.08% | 16.32% | 22.15% |
| **BERT base uncased** | 40.23% | 35.08% | 22.70% | 25.34% | 16.91% |
| **RoBERTa** | 39.05% | 58.94% | 23.01% | 30.56% | 27.13% |
| **XLM-RoBERTa** | 68.78% | 70.76% | 15.72% | 11.94% | 18.13% |

Table 5.6: Average top10 accuracy for every model and subset. For the closed interval subset, the results are from predicting Start&End years of the interval masking them separately.

As table 5.6 shows, the XLM-RoBERTa model has a high performance for the Closed Interval subset predictions, having around 70% accuracy top10 for both intervals duration and

the start and end years of the interval predictions. In the predictions of the start and end years of the interval, the DistilBERT base uncased and BERT base cased models also have similar accuracy values as the XLM-RoBERTa. In contrast, XLM-RoBERTa is clearly the model with the best performance for the duration predictions.

For the Open Interval subsets, the XLM-RoBERTa model is the worst. In this case, the DistilBERT base uncased is also one of the best for the Left Open and Right Open Interval subsets, with 28.76% and 29.8% accuracy values, respectively. For the Left Open Interval subset, the RoBERTa model has the best performance with 30.56% of accuracy.

Finally, for the Year Event subset, the behaviour of the models in terms of accuracy performance is similar to the Open Interval subsets. In this case, DistilBERT base uncased and RoBERTa models are also the best, with 26.26% and 27.13% top10 accuracy values, respectively.

In general, as aforementioned, there are significant differences in the performance of each model between the subsets and the five models within the subsets. Within the same subset, the best model can have around twice or three times the accuracy as the worst one, as happens in the Closed Interval Duration or the Left Open Interval subsets, respectively. Within models, there are clear differences too. For instance, the XLM-RoBERTa model has the best performance in some subsets but the worst in others.

### 5.5.2 Robustness

The robustness of the models has been analyzed too. More specifically, it is analyzed in two ways. The first one is the overall robustness of the models. As explained in the previous section, in general, the models are not robust since their levels of accuracy differ a lot from one subset to another. However, some models with very different performance behaviours, such as the XLM-RoBERTa, but others that, even if their accuracy values differ among subsets, are one of the models with the best performance in all the subsets studied, such as the DistilBERT base uncased model. The second way to study the robustness of the models has been checking the randomness of the predictions. As explained previously, since the first ten guesses were retrieved for each prediction, the standard deviation of each prediction is calculated. Then, all the standard deviations of an observation predictions are averaged. These values are shown in the table 5.7.

| | **Closed Interval** | **Closed Interval Duration** | **Right Open Interval** | **Left Open Interval** | **Year Event** |
|---|---|---|---|---|---|
| **DistilBERT base uncased** | 9.06 | 9.17 | 38.60 | 50.60 | 9.39 |
| **BERT base cased** | 10.08 | 6.99 | 70.32 | 26.2 | 9.31 |
| **BERT base uncased** | 11.39 | 9.18 | 32.25 | 32.95 | 9.62 |
| **RoBERTa** | 17.98 | 7.54 | 18.99 | 23.19 | 8.81 |
| **XLM-RoBERTa** | 12.68 | 7.33 | 42.31 | 42.16 | 8.02 |

Table 5.7: Average standard deviation in years of all the predictions per observation for every model and subset. For the closed interval subset, the results are from predicting Start&End years of the interval masking them separately.

As it can be seen, the Interval Duration predictions are the ones with the lowest standard deviation in all models, followed by the Year Event subset. In the Interval Duration predictions, XLM-RoBERTa was the model with the best accuracy performance and is the second one with the lowest standard deviation, with 7.33 years. The most robust model for this subset is the BERT base cased with a standard deviation among its predictions of 6.99 years. In the Year Event subset, the results show no significant difference among the models, being all of them robust, since the minimum average standard deviation in the predictions is 8.02 years for the XLM-RoBERTa model and the maximum is 9.62 years for the BERT base uncased. On the other hand, the Open Interval subsets have worse results in terms of standard deviation in the predictions. In this case, the most robust model is the RoBERTa, having an average standard deviation of 18.99 and 23.19 years for the Right Open Interval and the Left Open Interval subsets, respectively.

The accuracy performance and robustness can also be analyzed together in a more detailed way. That is, check the standard deviation values for each accuracy interval. This analysis is helpful to understand if the different models studied guess the correct answer randomly or not. If the model guesses the correct answer not randomly, the observations with high accuracy are expected to have lower standard deviation values. This would mean that the model guessed the correct answer in a not random way since all the guesses are closer to each other. On the other hand, the correct answer is expected to be guessed randomly if the standard deviation of the high accuracy interval is also high. This would mean that the model tried several options very different from each other and, more randomly or luckily, guessed the correct one. This study has been done per each model and subset, and the results are shown in the figures 5.4, 5.5, 5.6, 5.7 and 5.8.

As mentioned previously in this chapter, it can be seen that the average standard deviation values for the Closed Interval, the Interval Duration, and the Year Event subsets are lower than the Open Interval ones. However, in the figures 5.4, 5.5, 5.6, 5.7 and 5.8. can be seen in a more detailed way which models work better and their standard deviation levels behaviour for the high accuracy values intervals. For instance, in the Closed Interval subset, the best model in terms of accuracy is the XLM-RoBERTa, as shown in the table 5.6, but in terms of standard deviation, it is the second worse (table 5.7). However, when looking at the graphs for this subset in figure 5.4, it can be seen that this bad overall standard deviation value is due to a high value in the lowest accuracy interval, and how in the highest accuracy interval, the average standard deviation value is the lowest one. This indicates that this model works well since the highest accuracy interval is the one with more observations (high accuracy). When the model predicts correctly, the standard deviation is the lowest. Therefore, when this model predicts correctly, it is not randomly.

In the Interval Duration subset, the behaviour of the average standard deviation per accuracy interval is similar among the five models. The best model in the highest accuracy interval is the DistilBERT base uncased. However, the model with more percentage of data in the highest accuracy interval is the XLM-RoBERTa, and the average standard deviation values in that

interval are not far from the DistilBERT base uncased ones.

For the Year Event subset, the average standard deviation values for all the accuracy levels among all the models are low. This means that in this subset, the models, even if they do not have great accuracy values, their predictions are close to each other. Thus, the models predict consistently close values for all the predictions. However, these low average standard deviation values can also be explained by the fact that in this subset, some of the templates used can lead to predicted values being regular words instead of years with more probability. For example, with the template *[HEAD][RAW_RELATION][TAIL] "in" [MASK]*, it is possible that some of the retrieved predictions from the models are a place instead of a year. This means that the predicted values being years can be less in amount; therefore, the number of years to calculate the standard deviation is also lower. This situation can also happens for some templates of the Closed Interval subset.

Finally, the Open Interval ones are the subsets in which all models work the worst in terms of randomness. One of the best models in terms of accuracy for the Left Open Interval and Right Open Interval subsets is the DistilBERT base uncased, as seen in table 5.6. However, in terms of standard deviation, they have bad performance in both subsets, meaning that the predictions of this model in these subsets are more random than in the others. The other model with good accuracy performance in this subsets is the RoBERTa. In terms of standard deviation, it improves the values of the DistilBERT base uncased, but they are still far from the values in other subsets.

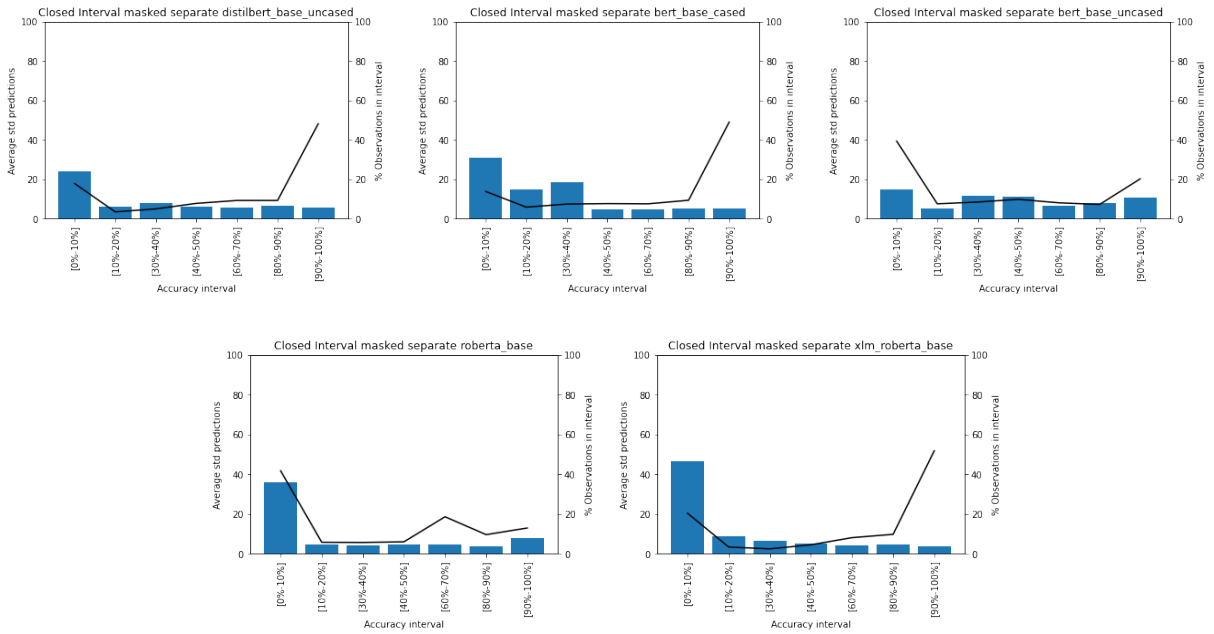


**Figure. 5.4:** Each plot shows the average standard deviation per top10 accuracy interval (represented by the blue bars) and the percentage of data each accuracy interval contains (represented by the black line) per each model in the Closed Interval subset.

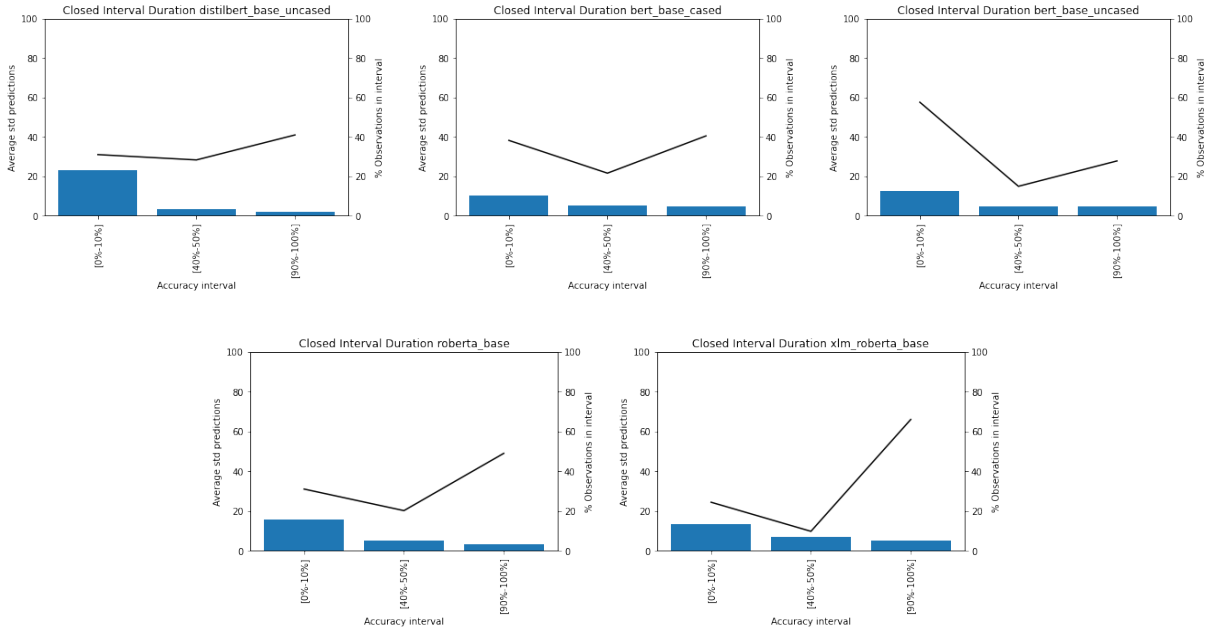


**Figure. 5.5:** Each plot shows the average standard deviation per top10 accuracy interval (represented by the blue bars) and the percentage of data each accuracy interval contains (represented by the black line) per each model in the Closed Interval Duration subset.

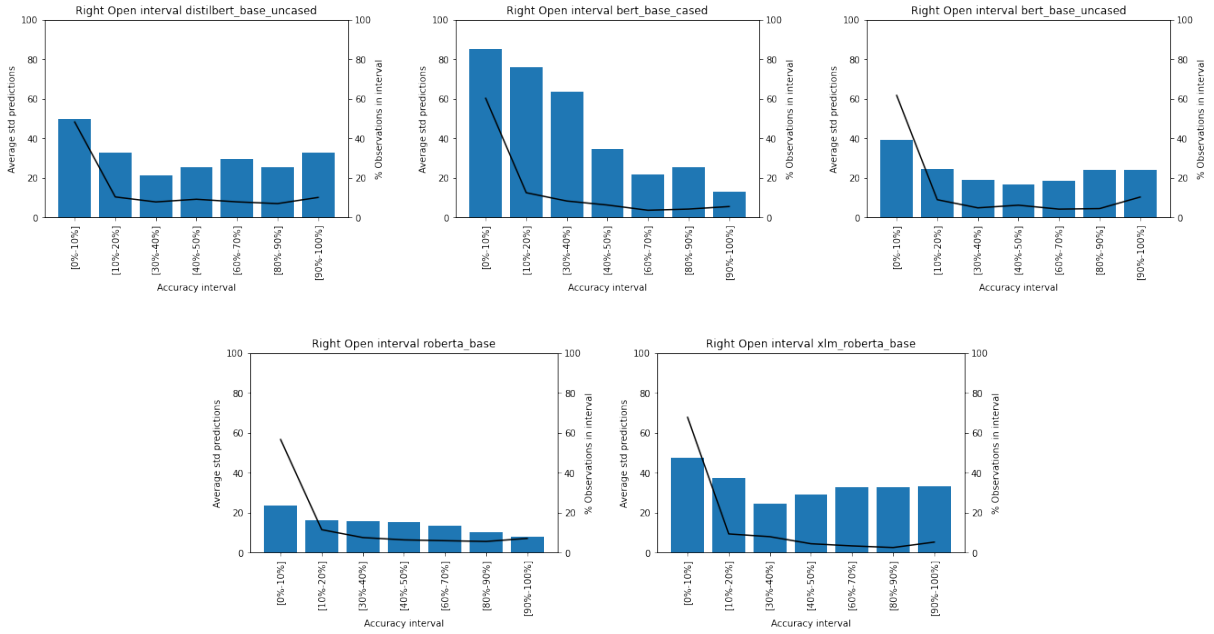


**Figure. 5.6:** Each plot shows the average standard deviation per top10 accuracy interval (represented by the blue bars) and the percentage of data each accuracy interval contains (represented by the black line) per each model in the Right Open Interval subset.

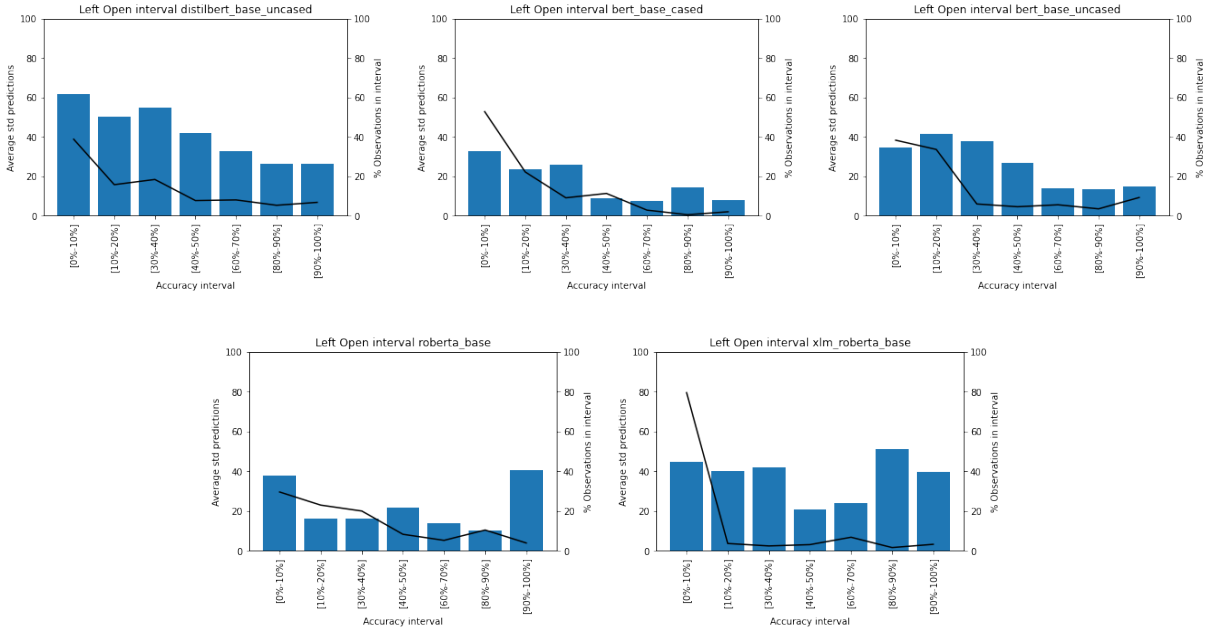


**Figure. 5.7:** Each plot shows the average standard deviation per top10 accuracy interval (represented by the blue bars) and the percentage of data each accuracy interval contains (represented by the black line) per each model in the Left Open Interval subset.

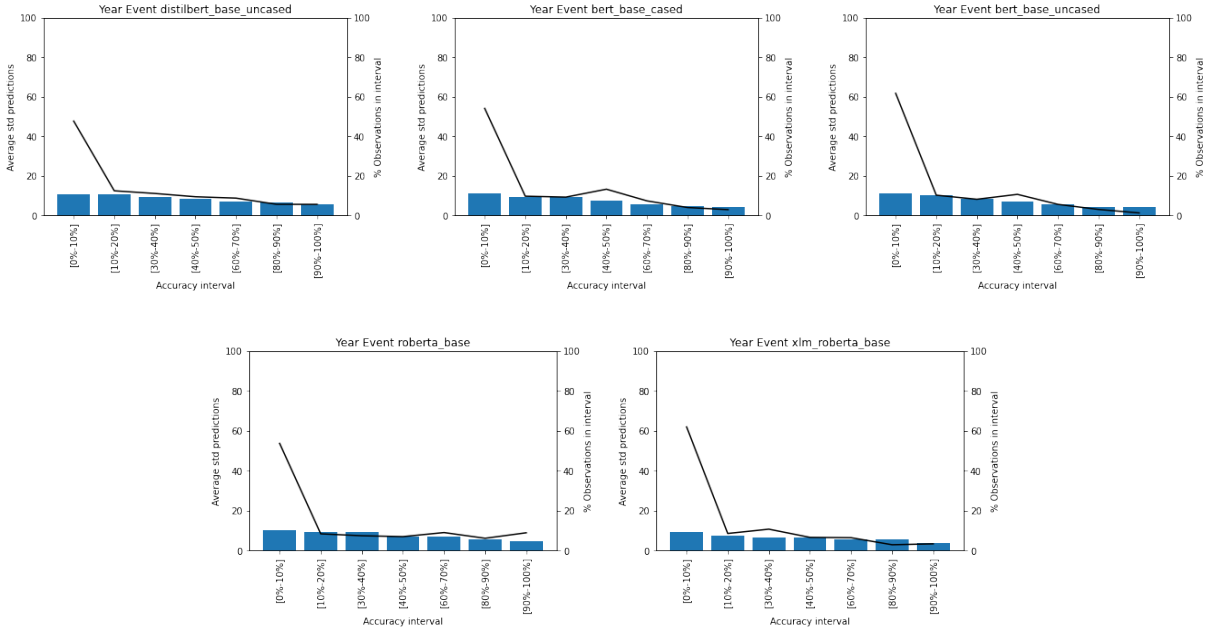


**Figure. 5.8:** Each plot shows the average standard deviation per top10 accuracy interval (represented by the blue bars) and the percentage of data each accuracy interval contains (represented by the black line) per each model in the Year Event subset.

## 5.6 Templates

Another analysis done in this thesis is related to the templates used. This analysis can be helpful in understanding if the context is important for the models to have better predictions. Also, to know if any specific method of creating the temporal part of the templates works better. Finally, it is useful to check if using the correct verbal tense and articles used in the sentence makes a difference for the models when predicting their answers.

Creating context for the templates is done by adding the description of the head and tail of the quadruple in the sentence. The context seems to be important for the models. The importance of context seems to vary depending on the subset. In the Closed Interval and Left Open Interval subsets, the best results are given by the templates with the context in them. More specifically, in the first of these two subsets, all of the six final subsets (the six best templates filtered out of the total) have context. For the Left Open Interval subset, 4 out of the six best filtered templates have context, and they are also the 4 with the best accuracy out of the 6. In the table 5.8, the best template for each subset can be seen as well as its accuracy top10.

| | Templates | Accuracy |
|---|---|---|
| **Closed Interval** | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] "between" [MASK] "and" [MASK] | 63.22% |
| **Closed Interval Duration** | [HEAD][TEMPLATE_RELATION][TAIL] "for" [MASK] "years" | 59.64% |
| **Right Open Interval** | [HEAD][TEMPLATE_RELATION][TAIL] "since" [MASK] | 24.44% |
| **Left Open Interval** | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] "finishing in the year" [MASK] | 41.94% |
| **Year Event** | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] "in the year" [MASK] | 27.67% |

Table 5.8: Best template per each subset in terms of accuracy top10. The accuracy top10 is averaged from all the models using each template.

On the other hand, the context doesn't seem to be necessary for the Interval Duration templates since none of the final filtered 2 with the best performance have context in them. Also, for the Right Open Interval subset, even if 4 out of the best templates have context in them, the 2 with the best performance are without context. For the Year Event subset, the context templates have better general results.

Regarding which temporal part of the template creation method is better, there seems to be a clear method for each subset. For the Closed Interval one, *"between [MASK] and [MASK]"* is the one used for the template with the best results, but also for the third-best one. For the Duration Interval subset, the *"for [MASK] years"* is the one used for the two final filtered templates. The Right Open Interval subset has similar results for all the filtered templates. However *"since"* is the method used for 3 of the final six templates. For the Left Open Interval one, even if the *"until [MASK]"* method is also used for 3 out of the six filtered templates, *"finishing in the year [MASK]"* is the one used for the best template and the difference from the second-best template is considerable, having 41.94% of accuracy for the best one and 26.08% for the second one. Moreover, also 2 out of the final six selected templates use this method. Finally, for the Year Event subset, the method using *"in the year [MASK]"* is the one with the best results, being used in the first three templates with the highest accuracy values.

On the other hand, when analyzing the relation part of the template, where the correct ver-

bal tense and the correct usage of articles are used in some templates, and just the raw relation for the others (with no article nor usage of the correct verbal tense), no significant differences can be seen. In this case, there are the templates using the template for the relations and the ones using the raw relations are mixed in the ranking.

Although all aforementioned characteristics of the template creation process are important, the final performance is given by the combination of them.

## 5.7 Reasoning

One of the purposes of this thesis is to understand if the language models used are able to reason. The duration templates were created to check whether they are able or not. Since the duration templates are created from a closed interval (with a start date and an end date), if the models can predict the number of years in the interval (which is the masked word in the duration templates), then it can be assumed that the pre-trained LMs can reason about time. As mentioned in the previous sections, the Interval Duration predictions are one of the best in terms of accuracy and robustness. This could mean that they can reason. However, not all the models have the same results. XLM-RoBERTa is the best model regarding the duration subset, and BERT base uncased is the worst. When looking at the two duration templates used, their results show no significant differences.

Overall, it can be said that XLM-RoBERTa is good at reasoning while BERT base uncased is not. With the others models used, it can not be concluded with certainty that they can reason.

# Chapter 6

# Discussion

In this thesis, it has been demonstrated that State-of-the-art pre-trained models for NLP can have acceptable results when predicting years under certain circumstances (e.g. predicting years of a closed interval masking them separately), but that a KG expansion using cloze statements by leveraging LM is still not reliable.

It is also worth mentioning the main limitations in the process of this thesis. The main drawback found in the process is the computational cost. Since the computational power available was limited, not all the models could be tried with all the dataset observations for all the templates proposed, but only for a 100 observations set for each subset of it. Also, the number of templates proposed was limited by this situation. With more powerful resources, more experiments could have been tried, and, therefore, more insights could have been learned.

It is worth mentioning the challenge that the time supposed. The tight time available to do the thesis (two months) is an aspect to consider. This fact limited some analyses, such as trying to understand whether the pre-trained LMs analyzed can reason or not or their robustness. An investigation was done for the two research questions related to these topics, although, with more time, a deeper analysis could have been done.

Moreover, it is also important to note the ethical issues the existing pre-trained LMs can carry with them. Specifically, all five pre-trained LMs used in this thesis were trained on Wikipedia, Common Crawl, or a dataset containing data from books. A consequence of this can be that, even if the datasets contain large amounts of data and the LMs perform well in general, the models can be biased. For instance, although Wikipedia has a lot of information, the information is not even since there is more information available regarding some topics than others. Therefore the models learning from this data can be not representative equally for everyone and everything. Common Crawl retrieves information from the Internet, but even if the amount of data it gathers is enormous, this data can be already biased from the source since everyone can put information on the Internet. The same happens with books. That means the models trained with these databases can have biased predictions since they are already biased

from the pre-training stage. We could all agree that since the information in the datasets used for pre-training the LMs is not entirely objective, they can contain fake news and misleading information. Thus, the models are also learning this information. Another issue that can arise regarding ethical concerns is the amount of data from each topic. Again, since everyone can participate from the information on the Internet, the topics covered in it can differ significantly regarding the amount of information available. This can lead to non-representative models and models biased towards the minorities in society. To conclude, in our society, there exist racism, sexism and many other important problems that the datasets used to pre-train the LMs can also capture. With this in mind, it would not be surprising that these LMs could reproduce these behaviours. Therefore, it is essential to keep in mind all of these ethical aspects when using these methods and think about whether they fit well the application that we want to give them or not because they can be beneficial, but also they could be dangerous at the same time.

On the other hand, in new projects with goals similar to this one, some things could be done to expand the findings. First of all, with more computational power available, more templates can be proposed by doing prompt engineering, and a bigger set of data could be selected to filter them. However, in a perfect scenario, all the templates proposed would be run for all the models, and no filtering process would be needed. Another further analysis that could be done to improve the reasoning insights is to make link predictions. That is, predicting the relation of a quadruple, given the head, tail, and temporal information. Related to this, the robustness analysis could also be improved in future works by making link predictions for several years (with the correct and wrong ones). For the robustness analysis, another interesting idea is to check the scores of the predictions retrieved by the models. A threshold could be set to understand how many predictions are over the threshold and, therefore, check the models' robustness. Moreover, a deeper analysis regarding the templates results could be done.

In a more general scope, further analysis to dig more into the ethical issues mentioned and the potential bias LMs can have could be done.

# Chapter 7

# Conclusions

The main aim of this thesis was to answer the three research questions proposed to be able to learn if pre-trained LMs can act as a KG and, therefore, be used for a KG expansion. In this chapter, the main insights are gathered and exposed.

The first research question aimed to know the performance of the existing pre-trained LMs on temporal cloze statements. The several analysis completed in this work showed that some of these models have acceptable results in predicting the years of an interval and its duration, with some models having around 70% of top10 accuracy. In contrast, the performance when predicting the years of an open interval or the year of an event is much lower, reaching a maximum of 30% top10 accuracy for the best model. Regarding which model is the best one in terms of accuracy, XLM-RoBERTa is the best for predicting the start and end year of an interval and its duration, while the RoBERTa and DistilBERT base uncased are the best ones when predicting the years of the open intervals and the event years. Also, DistilBERT is one of the models with the best results in all the subsets studied.

Another important insight found is that models' performance increases substantially when masking the years separately for the closed intervals. This can be explained due to the fact that, when masking separately, the models know the year that remains unmasked. Thus, it has more information.

An analysis was done to explore whether LMs can reason or not to answer the second research question. From this analysis, it can be concluded that, based on the models and methods used in this thesis, XLM-RoBERTa is the model that can reason the best, and BERT base uncased is the worse. Regarding the other models studied, it can not be assured that they are good at reasoning.

Finally, investigating the robustness of the pre-trained LMs is another aim this thesis had to answer the third research question. The findings were that the models as a general concept are not robust since they have different behaviour between and within them. However, when

checking the robustness of each one of them with the randomness of their predictions, it was found that XLM-RoBERTa, DistilBERT base uncased, and the BERT base cased can be considered a robust model for predicting the years of an interval, and the XLM-RoBERTa also for predicting an interval's duration.

From all the results gathered, it can be concluded that, for the time being, pre-trained LMs are not reliable enough to be used for an expansion from KG to a TKG.

# Bibliography


Asma Ben Abacha and Pierre Zweigenbaum. Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information processing & management*, 51(5):570–594, 2015.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 535–541. ACM, 2006. doi: 10.1145/1150402.1150464.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases. *CoRR*, abs/2106.09231, 2021. URL https://arxiv.org/abs/2106.09231.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL http://arxiv.org/abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Touseef Iqbal and Shaima Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences*, 2020.

Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. Temporal knowledge base completion: New algorithms and evaluation protocols. *CoRR*, abs/2005.05035, 2020. URL https://arxiv.org/abs/2005.05035.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, feb 2022. doi: 10.1109/tnnls.2021.3070843. URL https://doi.org/10.1109%2Ftnnls.2021.3070843.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *CoRR*, abs/1911.12543, 2019. URL http://arxiv.org/abs/1911.12543.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019. ISSN 2078-2489. doi: 10.3390/info10040150. URL https://www.mdpi.com/2078-2489/10/4/150.

Elizabeth D Liddy. Natural language processing. 2001.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases? *CoRR*, abs/1909.01066, 2019. URL http://arxiv.org/abs/1909.01066.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. TIMEDIAL: temporal commonsense reasoning in dialog. *CoRR*, abs/2106.04571, 2021. URL https://arxiv.org/abs/2106.04571.

Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. *CoRR*, abs/2102.07350, 2021. URL https://arxiv.org/abs/2102.07350.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. Question answering over temporal knowledge graphs. *CoRR*, abs/2106.01515, 2021. URL https://arxiv.org/abs/2106.01515.

Wilson L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953. doi: 10.1177/107769905303000401. URL https://doi.org/10.1177/107769905303000401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian M. Suchanek. Machine knowledge: Creation and curation of comprehensive knowledge bases. *CoRR*, abs/2009.11564, 2020. URL https://arxiv.org/abs/2009.11564.

Jiajun Zhang and Chengqing Zong. Deep neural networks in machine translation: An overview. *IEEE Intell. Syst.*, 30(5):16–25, 2015. doi: 10.1109/MIS.2015.69.

# Appendix-Templates

| Subset | Templates |
|---|---|
| Closed Interval | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “from the year” [MASK] “to the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “from the year” [MASK] “to the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “from the year” [MASK] “to the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “from the year” [MASK] “to the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “since” [MASK] “until” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “since” [MASK] “until” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “since” [MASK] “until” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “since” [MASK] “until” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “from” [MASK] “to” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “from” [MASK] “to” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “from” [MASK] “to” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “from” [MASK] “to” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “between the years” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “between the years” [MASK] “and” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “between the years” [MASK] “and” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “between the years” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “between the year” [MASK] “and the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “between the year” [MASK] “and the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “between the year” [MASK] “and the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “between the year” [MASK] “and the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “between” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “between” [MASK] “and” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “between” [MASK] “and” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “between” [MASK] “and” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “since the year” [MASK] “until the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “since the year” [MASK] “until the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “since the year” [MASK] “until the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “since the year” [MASK] “until the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “starting from the year” [MASK] “and finishing in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “starting from the year” [MASK] “and finishing in the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “starting from the year” [MASK] “and finishing in the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “starting from the year” [MASK] “and finishing in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “starting from” [MASK] “and finishing in” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “starting from” [MASK] “and finishing in” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “starting from” [MASK] “and finishing in” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “starting from” [MASK] “and finishing in” [MASK] |
| Closed Interval Duration | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “for” [MASK] “years” |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “for” [MASK] “years” |
| | [HEAD][RAW_RELATION][TAIL] “for” [MASK] “years” |
| | [HEAD][TEMPLATE_RELATION][TAIL] “for” [MASK] “years” |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “during” [MASK] “years” |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “during” [MASK] “years” |
| | [HEAD][RAW_RELATION][TAIL] “during” [MASK] “years” |
| | [HEAD][TEMPLATE_RELATION][TAIL] “during” [MASK] “years” |
| Open Interval | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “since”/”until” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “since”/”until” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “since”/”until” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “since”/”until” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “since the year”/”until the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “since the year”/”until the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “since the year”/”until the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “since the year”/”until the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “starting from the year”/”finishing in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “starting from the year”/”finishing in the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “starting from the year”/”finishing in the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “starting from the year”/”finishing in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “starting from”/”finishing in” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “starting from”/”finishing in” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “starting from”/”finishing in” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “starting from”/”finishing in” [MASK] |
| Year Event | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “in” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “in” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “in” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “in” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][RAW_RELATION][TAIL][DESCRIPTION_TAIL] “in the year” [MASK] |
| | [HEAD][DESCRIPTION_HEAD][TEMPLATE_RELATION][TAIL][DESCRIPTION_TAIL] “in the year” [MASK] |
| | [HEAD][RAW_RELATION][TAIL] “in the year” [MASK] |
| | [HEAD][TEMPLATE_RELATION][TAIL] “in the year” [MASK] |

Table 1: All templates proposed for each subset. For the Open Interval subset, the first version of the temporal part of the sentence (e.g. ”since”) is for the templates of the Right Open Interval. The Left Open Interval has the same templates using the second version of the temporal part of the sentence (e.g. ”until”).

# Appendix-Models

| | Acc top1 | Acc top3 | Acc top5 | Acc top10 |
|---|---|---|---|---|
| **DistilBERT base uncased** | 13.59% | 35.46% | 61.06% | 86.16% |
| **BERT base cased** | 12.78% | 32.97% | 52.70% | 77.56% |
| **BERT base uncased** | 11.59% | 24.24% | 38.31% | 62.99% |
| **RoBERTa** | 12.22% | 29.14% | 47.28% | 74.36% |
| **XLM-RoBERTa** | 17.19% | 43.47% | 60.02% | 79.13% |

Table 2: The four accuracy measures calculated per model for predicting the end year of the Closed Interval subset masking separately.

| | Acc top1 | Acc top3 | Acc top5 | Acc top10 |
|---|---|---|---|---|
| **DistilBERT base uncased** | 11.92% | 28.09% | 45.45% | 74.34% |
| **BERT base cased** | 12.38% | 34.44% | 53.57% | 80.74% |
| **BERT base uncased** | 9.20% | 22.18% | 33.04% | 54.25% |
| **RoBERTa** | 7.09% | 17.11% | 26.58% | 44.37% |
| **XLM-RoBERTa** | 11.39% | 30.07% | 49.82% | 75.67% |

Table 3: The four accuracy measures calculated per model for predicting the start year of the Closed Interval subset masking separately.

| | Acc top1 | Acc top3 | Acc top5 | Acc top10 |
|---|---|---|---|---|
| **DistilBERT base uncased** | 3.21% | 15.97% | 33.36% | 67.81% |
| **BERT base cased** | 4.35% | 17.43% | 36.72% | 68.78% |
| **BERT base uncased** | 2.99% | 10.80% | 19.99% | 40.23% |
| **RoBERTa** | 2.77% | 10.52% | 19.78% | 39.05% |
| **XLM-RoBERTa** | 3.17% | 17.61% | 37.57% | 68.78% |

Table 4: The four accuracy measures calculated per model for predicting both the start and end years of the Closed Interval subset masking separately.

| | **Acc top1** | **Acc top3** | **Acc top5** | **Acc top10** |
|---|---|---|---|---|
| **DistilBERT base uncased** | 4.95% | 10.68% | 15.83% | 27.71% |
| **BERT base cased** | 5.51% | 13.59% | 20.75% | 36.03% |
| **BERT base uncased** | 6.15% | 14.19% | 20.77% | 34.34% |
| **RoBERTa** | 7.37% | 15.25% | 22.62% | 39.58% |
| **XLM-RoBERTa** | 6.69% | 13.98% | 20.76% | 36.97% |

Table 5: The four accuracy measures calculated per model for predicting the end year of the Closed Interval subset masking together.

| | **Acc top1** | **Acc top3** | **Acc top5** | **Acc top10** |
|---|---|---|---|---|
| **DistilBERT base uncased** | 4.29% | 9.78% | 13.94% | 21.52% |
| **BERT base cased** | 5.32% | 10.44% | 15.30% | 25.86% |
| **BERT base uncased** | 4.07% | 8.86% | 12.84% | 21.89% |
| **RoBERTa** | 3.59% | 7.66% | 11.56% | 20.82% |
| **XLM-RoBERTa** | 4.76% | 10.78% | 15.52% | 27.04% |

Table 6: The four accuracy measures calculated per model for predicting the start year of the Closed Interval subset masking together.

| | **Acc top1** | **Acc top3** | **Acc top5** | **Acc top10** |
|---|---|---|---|---|
| **DistilBERT base uncased** | 1.12% | 3.27% | 5.25% | 11.24% |
| **BERT base cased** | 1.23% | 2.97% | 5.19% | 13.20% |
| **BERT base uncased** | 0.99% | 2.99% | 4.98% | 10.94% |
| **RoBERTa** | 1.07% | 2.05% | 3.28% | 9.04% |
| **XLM-RoBERTa** | 0.74% | 1.96% | 3.92% | 11.88% |

Table 7: The four accuracy measures calculated per model for predicting both the start and end years of the Closed Interval subset masking together.

| | **Acc top1** | **Acc top3** | **Acc top5** | **Acc top10** |
|---|---|---|---|---|
| **DistilBERT base uncased** | 5.64% | 13.29% | 22.19% | 54.95% |
| **BERT base cased** | 3.29% | 11.44% | 22.05% | 51.12% |
| **BERT base uncased** | 6.78% | 14.52% | 21.36% | 35.08% |
| **RoBERTa** | 5.28% | 22.95% | 35.48% | 58.94% |
| **XLM-RoBERTa** | 8.31% | 28.43% | 47.92% | 70.76% |

Table 8: The four accuracy measures calculated per model for predicting the duration of the Closed Interval subset.

| | **Acc top1** | **Acc top3** | **Acc top5** | **Acc top10** |
|---|---|---|---|---|
| **DistilBERT base uncased** | 3.53% | 10.93% | 16.29% | 29.80% |
| **BERT base cased** | 1.49% | 6.01% | 9.83% | 19.08% |
| **BERT base uncased** | 2.67% | 7.48% | 12.18% | 22.70% |
| **RoBERTa** | 2.87% | 8.09% | 12.17% | 23.01% |
| **XLM-RoBERTa** | 1.41% | 5.31% | 8.18% | 15.72% |

Table 9: The four accuracy measures calculated per model for predicting year of the Right Open Interval subset.

| | Acc top1 | Acc top3 | Acc top5 | Acc top10 |
|---|---|---|---|---|
| **DistilBERT base uncased** | 8.40% | 17.90% | 22.65% | 28.76% |
| **BERT base cased** | 1.12% | 3.32% | 5.48% | 16.32% |
| **BERT base uncased** | 4.01% | 9.71% | 16.58% | 25.34% |
| **RoBERTa** | 4.69% | 13.78% | 20.76% | 30.56% |
| **XLM-RoBERTa** | 3.25% | 6.95% | 8.62% | 11.94% |

Table 10: The four accuracy measures calculated per model for predicting year of the Left Open Interval subset.

| | Acc top1 | Acc top3 | Acc top5 | Acc top10 |
|---|---|---|---|---|
| **DistilBERT base uncased** | 3.34% | 9.73% | 14.83% | 26.26% |
| **BERT base cased** | 1.78% | 6.51% | 11.18% | 22.15% |
| **BERT base uncased** | 1.48% | 5.12% | 8.69% | 16.91% |
| **RoBERTa** | 3.29% | 8.82% | 13.72% | 27.13% |
| **XLM-RoBERTa** | 0.82% | 4.26% | 8.06% | 18.13% |

Table 11: The four accuracy measures calculated per model for predicting year of the Event Year subset.