



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

## Prompting as Probing: Using Language Models for Knowledge Base Construction

### Citation for published version:

Alivanistos, D, Santamaría, SB, Cochez, M, Kalo, JC, van Krieken, E & Thanapalasingam, T 2022, Prompting as Probing: Using Language Models for Knowledge Base Construction. in S Singhanía, T-P Nguyen & S Razniewski (eds), *LM-KBC 2022 Knowledge Base Construction from Pre-trained Language Models 2022: Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022 co-located with the 21st International Semantic Web Conference (ISWC2022)*. vol. 3274, CEUR Workshop Proceedings, CEUR-WS.org, pp. 11-34, Knowledge Base Construction from Pre-trained Language Models 2022, 1/10/22. <<https://ceur-ws.org/Vol-3274/paper2.pdf>>

### Link:

[Link to publication record in Edinburgh Research Explorer](#)

### Document Version:

Publisher's PDF, also known as Version of record

### Published In:

LM-KBC 2022 Knowledge Base Construction from Pre-trained Language Models 2022

### General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

### Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Prompting as Probing: Using Language Models for Knowledge Base Construction

Dimitrios Alivanistos<sup>1,2,4</sup>, Selene Báez Santamaría<sup>1,2</sup>, Michael Cochez<sup>1,2,4</sup>,  
Jan-Christoph Kalo<sup>1,2,5</sup>, Emile van Krieken<sup>1,2</sup> and Thiviyan Thanapalasingam<sup>1,2,3</sup>

<sup>1</sup>Authors are listed in alphabetical order to denote equal contributions.

<sup>2</sup>Vrije Universiteit Amsterdam

<sup>3</sup>Universiteit van Amsterdam

<sup>4</sup>Discovery Lab, Elsevier, The Netherlands

<sup>5</sup>DReaMS Lab, Huawei, The Netherlands

## Abstract

Language Models (LMs) have proven to be useful in various downstream applications, such as summarisation, translation, question answering and text classification. LMs are becoming increasingly important tools in Artificial Intelligence, because of the vast quantity of information they can store. In this work, we present ProP (Prompting as Probing), which utilizes GPT-3, a large Language Model originally proposed by OpenAI in 2020, to perform the task of Knowledge Base Construction (KBC). ProP implements a multi-step approach that combines a variety of prompting techniques to achieve this. Our results show that manual prompt curation is essential, that the LM must be encouraged to give answer sets of variable lengths, in particular including empty answer sets, that true/false questions are a useful device to increase precision on suggestions generated by the LM, that the size of the LM is a crucial factor, and that a dictionary of entity aliases improves the LM score. Our evaluation study indicates that these proposed techniques can substantially enhance the quality of the final predictions: ProP won track 2 of the LM-KBC competition, outperforming the baseline by 36.4 percentage points. Our implementation is available on <https://github.com/HEmile/iswc-challenge>.

## 1. Introduction

Language Models (LMs) have been at the center of attention, presented as a recent success story of Artificial Intelligence. LMs have shown great promise across a wide range of domains in a variety of different tasks, such as Text Classification [1], Financial Sentiment Analysis [2], and Protein Binding Site Prediction [3]). In recent years, prompt engineering for LMs has become a research field in itself, with a plethora of papers working on LM understanding (e.g. [4]).

Natural Language Processing (NLP) researchers have recently investigated whether LMs could potentially be used as Knowledge Bases, by querying for particular information. In Petroni *et al.* [5], the LAMA dataset for probing relational facts from Wikidata in LMs was presented.

---


*LM-KBC'22: Knowledge Base Construction from Pre-trained Language Models, Challenge at ISWC 2022*

✉ d.alivanistos@vu.nl (D. Alivanistos); s.baezsantamaria@vu.nl (S. B. Santamaría); m.cochez@vu.nl (M. Cochez); j.c.kalo@vu.nl (J. Kalo); e.van.krieken@vu.nl (E. v. Krieken); t.thanapalasingam@uva.nl (T. Thanapalasingam)

🌐 <https://dimitrisalivas.github.io/> (D. Alivanistos); <https://selbaez.github.io/> (S. B. Santamaría); <https://www.cochez.nl/> (M. Cochez); <https://research.vu.nl/en/persons/jan-christoph-kalo> (J. Kalo); <https://emilevankrieken.com/> (E. v. Krieken); <https://thiviyansingam.com/> (T. Thanapalasingam)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The authors show that the masked LM BERT can complete Wikidata facts with a precision of around 32%. Several follow-up papers have pushed this number to almost 50% [6, 7]. While the prediction quality on LAMA is promising, others have argued that LMs should not be used as knowledge graphs, but rather to support the augmentation and curation of KGs [8].

In this paper, we describe ProP, the system we implemented for the “Knowledge Base Construction from Pre-trained Language Models” challenge at ISWC 2022<sup>1</sup>. The task is to predict possible objects of a triple, where the subject and relation are given. For example, given the string "Ronnie James Dio" as the subject and the relation *PersonInstrument*, an LM needs to predict the answers "bass guitar" or "guitar", and "trumpet", as the objects of the triple. In contrast to the LAMA dataset [5]), the LM-KB challenge dataset contains questions for which there are no answers or where multiple answers are valid. Bakel *et al.* argue for precision and recall metrics over the common ranking-based metrics for query answering. [9]. Since the LM-KBC dataset requires predicting a set of answers and in some cases even empty answer sets, we choose precision and recall metrics to evaluate our system.

ProP uses the GPT-3 model, a large LM proposed by OpenAI in 2020[10]. For each relation type, we engineered prompts that, when given to the language model, probe it to respond with the set of objects for that relation. The components of ProP can be divided into two categories: **prompt generation** focuses on generating the correct prompts that yield the desired answers for the questions in the LM-KBC dataset, and **post-processing** aims to enhance the quality of the predictions.

## 2. Related Work

The development of large pre-trained Language Models (LMs) has led to substantial improvements in NLP research. It was shown that extensive pre-training on large text corpora encodes large amounts of linguistic and factual knowledge into a language model that can help to improve the performance on various downstream tasks (see [11] for a recent overview).

**LM as KG** Petroni et al. [5], and later others [8], asked the question to what extent language models can replace or support the creation and curation of knowledge graphs. Petroni et al. proposed the LAMA dataset for probing relational knowledge in language models by using masked language models to complete cloze-style sentences. As an example, the language model BERT can complete the sentence “Paris is the capital of [MASK]” with the word “France”. In this case, it is assumed that the model *knows* about, or can predict, the triple (Paris, capitalOf, France).

While the original paper relied on manually designed prompts for probing the language model, various follow-up works have shown the superiority of automatically learning prompts. Methods can mine prompts from large text corpora and pick the best prompts by applying them to a training dataset as demonstrated in [12, 13]. Prompts can also be directly learned via backpropagation: BERTESE [14] and AutoPrompt [15] show how prompts can be learned to improve the performance on LAMA. The probing performance can be pushed even further by either directly learning continuous embeddings for the prompts [7, 16] or by directly fine-tuning

---

<sup>1</sup>LM-KBC, <https://lm-kbc.github.io/>. This work is a submission in the open track (Track 2) in which LMs of any sizes can be used

the LM on the training data [17]. Similar to our work, in FewShot-LAMA [18] few-shot learning on the original LAMA dataset is evaluated. The authors show that a combination of few-shot examples with learned prompts achieves the best probing results.

Since the publication of the LAMA dataset, a large variety of other probing datasets for factual knowledge in LMs have been created. LAMA-UHN is a more difficult version of LAMA [19]. TimeLAMA adds a time component to facts that the model is probed for [20]. Furthermore, BioLAMA [21] and MedLAMA [22] are domain-specific probing datasets for biological and medical knowledge.

Most existing approaches have in common that they only probe the language models for entities with a label consisting of a single token from the language model’s vocabulary. Thus, the prediction of long, complex entity names is mostly unexplored. Furthermore, most existing works have only asked language models to complete a triple with a single prediction, even though some triples might actually allow for multiple possible predictions. Both these aspects substantially change the challenge of knowledge graph construction.

**LM for KG** While probing language models has been heavily studied in the NLP community, the idea of using language models to support knowledge graph curation is not sufficiently studied [8]. Some works have shown how LMs in combination with knowledge graphs can be used to complete query results [23]. Other works have looked into how to use language models to identify errors in knowledge graphs [24], or have studied how to weight KG triples from ConceptNet with language models to measure semantic similarity. Biswas et al. [25] have shown that language models can be used to perform entity typing by predicting the class using language models.

KG-BERT is a system that is most similar to what is required for the KBC Challenge [26]. A standard BERT model is trained on serialized knowledge graph data to perform link prediction on standard link prediction benchmark datasets. KG-BERT’s performance on link prediction is comparable to many state-of-the-art systems that use knowledge graph embeddings for this task.

**Similar tasks** This KBC Challenge task is similar to Factual Q&A with Language models, where the goal is to respond to questions that fall outside the scope of a knowledge base. This shared task differs in that the responses need to include 0 to k answers. Moreover, in the shared task, the factual questions are generated from triples, thus including variation in how a triple might be phrased as a question.

### 3. The LM-KBC Challenge

The LM-KBC dataset contains triples of the form  $(s, p, O)$ , where  $s$  is the textual representation of the *subject*,  $p$  is one of 12 different predicates and  $O$  the (possibly empty) set of textual representations of *object* entities for prediction. The subjects and objects are of various types. After learning from a training set of such triples, given a new subject and one of the known relations, the task is to predict the complete set of objects.

### 3.1. The LM-KBC Dataset

For each of the 12 relations, the number of unique subject-entities in the train, dev, and test sets are 100, 50, and 50 respectively. We include detailed distributions of the cardinality (the number of object-entities) for each relation type (Appendix, Figures 2 and 3). Table 4 in the Appendix shows the aggregated statistics about the number of object-entities, as well as the number of alternative labels per object-entities in the development set. Certain relation types have a much higher average cardinality (e.g. *PersonProfession*=7.42 or *StateSharesBorderState*=5.6) than others (e.g. *CompanyParentOrganization*=0.32, *PersonPlaceOfDeath*=0.50). We also note that only five of the relations allow for empty answer sets. For example, relations associated with a person’s death (*PersonPlaceOfDeath* and *PersonCauseOfDeath*) often contain empty answer sets, because many persons in the dataset are still alive. In these cases, a models needs to be able to predict empty answer sets correctly.

### 3.2. The Baseline

The baseline model is a masking-based approach that uses the popular BERT model [27] in 3 variants (*base-cased*, *large-cased* and *RoBERTa* [28]). The BERT model is tasked with doing prompt completion to predict object-entities for a given subject-relation pair. The prompts used by the baseline approach have been customised for the different relation types.

## 4. Our Method

Previous studies have shown that prompt formulation plays a critical role in the performance of LMs on downstream applications [29, 4] and this also applies to our work. We investigated different prompting approaches using few-shot learning with GPT-3 [10] using OpenAI’s API <sup>2</sup>. In this Section, we first describe how we generate the prompts (prompt generation phase), and followed by how we utilise different components in our pipeline to further fine-tune the prompts for an enhanced performance (post-processing phase).

### 4.1. Prompt Generation

For each relation in the dataset, we manually curate a set of prompt templates consisting of four representative examples<sup>3</sup> selected from the training split. We use these relation-specific templates to generate prompts for every subject entity by replacing a placeholder token in the final line of the template with the subject entity of interest. We task GPT-3 to complete the prompts, and evaluate the *completions* and compute the macro-precision, macro-accuracy and macro-F1 score for each relation.

---

<sup>2</sup><https://openai.com/api> (temperature=0, max\_tokens=100, top\_p=1, frequency\_penalty=0, presence\_penalty=0, logprobs=1)

<sup>3</sup>This is an arbitrary number of training examples. Since few-shot learners are efficient at learning from a handful of examples [10], including an excessive number of training examples may not necessarily lead to any improved precision or recall. Therefore, we did not study the effects of varying the number of training examples in the prompts.

We ensured that the training examples for few-shot learning included all of the following: (i) questions with answer sets of variable lengths to inform the LM that it can generate multiple answers; (ii) questions with empty answer sets to ensure that the LM returns nothing when there is no correct answer to a given question; (iii) a fixed question-answer order, where we provide the question and then immediately the answer so that the LM learns the style of the format, and (iv) the answer set formatted as a list to ensure we can efficiently parse the answer set from the LM. We did not study the order of these examples, but hypothesize that this is not hugely important to GPT-3 as it can handle long-range dependencies well.

We formulate the questions either in natural language or in the form of a triple. We hand-designed the natural language questions and did not compare them in a structured manner to alternatives. However, we tried out several variations in OpenAI GPT-3 playground to get an intuition of what style of questions are effective. We found those are usually shorter and simpler. We include the prompt templates used in Section 7.2 of the Appendix. In our work, we investigate the use of both prompting styles and compare natural language prompts with triple-based prompts for the different relations.

## 4.2. Empty Answer sets

There are some questions in the dataset for which the correct answer is an empty answer set. For instance, there are no countries that share borders with Fiji. The empty answer set can be represented as either an empty list [ ], or as a string within a list [ 'None' ]. We have observed (see Table 2) that the way that empty sets are represented affects the precision and recall of our approach. Allowing the explicit answer 'None' encourages the LM to exclude answers that it is uncertain about.

## 4.3. Fact probing

Our initial results indicated that the recall of our approach was high, but the precision for certain relations was low. Therefore, we add a post-processing step called *fact probing* in ProP's pipeline. We use fact probing to ask the LM to probe whether each completion proposed by the LM in the previous step is correct. Inspired by maieutic prompting [30], we create a simple prompt, where we translate each predicted completion into a natural language fact. Then we ask the LM to predict whether this fact is true or false. One example of a fact-probing prompt is *Niger neighbours Libya TRUE* for the *CountryBordersWithCountry* relation. We ensure that the LM only predicts either TRUE or FALSE by adding a true and a false example to the prompt.

## 5. Results

In this Section, we analyse the ProP pipeline we built for generating prompts and evaluate the contribution of each component. We explain how we combine the best-performing components to yield a prediction that obtains a high macro F1-score on the test split.

## 5.1. Prompt-Fine Tuning

### 5.1.1. Natural Language vs Triple-based Prompts

Table 1 shows the quality of the predictions for the natural language prompts and triple-based prompts. We note that on F1, the performance between these two prompt styles is mixed, with F1 being higher on the triple-style prompts in only seven out of twelve cases. Unpacking the F1 into recall and precision shows that the triple-style prompts yield higher precision, while natural language prompts yield higher recall. Overall, the triple-style prompts do yield a higher F1 when averaged over each relation. Our intuition is that natural language prompts contain certain words that badly influence the precision of the predictions. It is, however, difficult to study this systematically as the enumeration of word combinations in the prompts is very large. Triple-based prompts circumvent this problem because they only contain the relevant terms in the prompts for the subject entities and relations that are required to predict the object entities.

**Table 1**

Precision, Recall, and F1-score for predictions generated using natural language and triple-based prompts across the different relations. Results are on the dev dataset. Values rounded up to three decimal places. Best scores are in **bold**.

Relation Type	Precision		Recall		F1	
Method	Triple	Natural Language	Triple	Natural Language	Triple	Natural Language
ChemicalCompoundElement	<b>0.976</b>	0.895	<b>0.919</b>	0.885	<b>0.940</b>	0.884
CompanyParentOrganization	<b>0.587</b>	0.385	<b>0.600</b>	0.400	<b>0.590</b>	0.388
CountryBordersWithCountry	<b>0.865</b>	0.809	0.733	<b>0.800</b>	0.766	<b>0.785</b>
CountryOfficialLanguage	<b>0.933</b>	0.798	0.810	<b>0.882</b>	<b>0.833</b>	0.785
PersonCauseOfDeath	<b>0.560</b>	0.500	<b>0.550</b>	0.500	<b>0.553</b>	0.500
PersonEmployer	0.261	<b>0.273</b>	0.267	<b>0.323</b>	0.226	<b>0.262</b>
PersonInstrument	<b>0.547</b>	0.489	<b>0.508</b>	0.458	<b>0.502</b>	0.446
PersonLanguage	<b>0.840</b>	0.750	0.894	<b>0.932</b>	<b>0.827</b>	0.793
PersonPlaceOfDeath	0.820	<b>0.840</b>	0.820	<b>0.840</b>	0.820	<b>0.840</b>
PersonProfession	0.669	<b>0.713</b>	0.527	<b>0.535</b>	0.556	<b>0.581</b>
RiverBasinsCountry	<b>0.845</b>	0.820	<b>0.868</b>	0.863	<b>0.832</b>	0.822
StateSharesBorderState	0.587	<b>0.628</b>	0.407	<b>0.462</b>	0.472	<b>0.522</b>
<b>Average over all relations</b>	<b>0.707</b>	0.658	<b>0.658</b>	0.657	<b>0.660</b>	0.634

### 5.1.2. Empty vs None

As explained in Section 3.1, five out of twelve relations allow for empty answer sets. We experiment with the different ways to represent such empty answer sets, and Table 2 shows the results. Three relations get a performance boost when prompted with ‘NONE’ (*CompanyParentOrganization*, *PersonCauseOfDeath*, *PersonInstrument*), while the other two relations perform better when using empty lists (*CountryBordersWithCountry*, *PersonPlaceOfDeath*). For subsequent experiments, we modified the prompt of each relation to use the best performant representation.

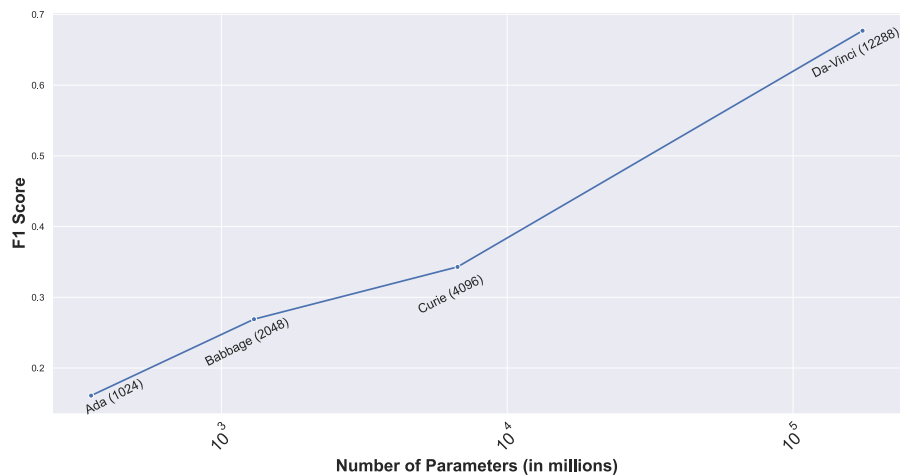
**Table 2**

Precision, recall, and F1-score for the Davinci model across the different relations. Results are on the dev dataset. We only include those relations which require empty answer sets. Values are rounded up to three decimal places. Best scores are in **bold**.

Relation Type	Precision		Recall		F1	
Method	Empty	None	Empty	None	Empty	None
CompanyParentOrganization	0.587	<b>0.767</b>	0.600	<b>0.780</b>	0.590	<b>0.770</b>
CountryBordersWithCountry	<b>0.865</b>	0.826	<b>0.733</b>	0.719	<b>0.766</b>	0.749
PersonCauseOfDeath	0.560	<b>0.600</b>	0.550	<b>0.590</b>	0.553	<b>0.593</b>
PersonInstrument	0.547	<b>0.600</b>	0.508	<b>0.561</b>	0.502	<b>0.568</b>
PersonPlaceOfDeath	<b>0.820</b>	0.780	<b>0.820</b>	0.780	<b>0.820</b>	0.780
<b>Average over all relations</b>	0.685	<b>0.697</b>	0.674	<b>0.685</b>	0.657	<b>0.669</b>

### 5.1.3. Language Model Size

Usually, the size of LMs is measured by the number of learnable parameters in the model. However, the OpenAI API does not quantify the number of total parameters but only the size of the embedding dimensions for the tokens<sup>4</sup>. We assume there is a positive correlation between the token dimension size and the total number of GPT-3 parameters. Figure 1 shows our results, and we can see that as the language model size increases, the F1-score also increases. This shows that a larger LM gives better performance on KBC.



**Figure 1:** The F1-scores of GPT-3 models with different number of parameters. In brackets, the embedding dimension for the model. We observe an almost analogous increase between size and performance.

<sup>4</sup><https://beta.openai.com/docs/models/gpt-3>



## 5.2. Post-Processing Predictions

Up to this point, we have discussed how we generate the optimal prompts for the different relation types. Once the LM produces the completions using these optimal prompt techniques, we can employ two additional steps to enhance the precision and recall of our predictions. Table 3 shows the results of including fact probing and entity aliases in our system.

### 5.2.1. Fact probing

We found that fact probing has a different impact on different relation types. This difference could stem from the cardinalities of the relation types. For example, the relation *PersonPlaceOfDeath*, which has only one correct answer, should show a larger improvement than *State Borders*, which has a higher cardinality. We found that fact probing helped to boost the predictions of five relations (*CompanyParentOrganization*, *CountryOfficialLanguage*, *PersonCauseOfDeath*, *PersonInstrument*, *PersonLanguage*). We only apply fact probing to these relations. On the dev set, the precision of fact probing is 0.737, and 0.608 among the predictions removed by fact probing. That is, in 60.8% fact probing filtered a prediction, it correctly removed a prediction that was not in the ground truth set.

**Table 3**

Precision, recall, and F1-score for predictions generated using different post-processing techniques on the development (dev) and test sets. We round up values to three decimal places. Best F1-scores are in **bold**.

Method	Precision	Recall	F1
BERT Baseline (dev)	0.349	0.295	0.309
BERT Baseline + fact probing (dev)	0.357	0.304	<b>0.317</b>
BERT Baseline + fact probing + alias (dev)	0.357	0.304	0.317
GPT3-Davinci (dev)	0.736	0.699	0.697
GPT3-Davinci + fact probing (dev)	0.741	0.692	0.698
GPT3-Davinci + fact probing + alias (dev)	0.755	0.709	<b>0.712</b>
GPT3-Davinci (test)	0.782	0.701	0.679
GPT3-Davinci + fact probing (test)	0.798	0.690	0.676
GPT3-Davinci + fact probing + alias (test)	0.813	0.704	<b>0.689</b>

### 5.2.2. Entity aliases

As we discussed in Section 3.1, the predictions from the language model are sometimes correct, but not according to what the gold standard expects. Whether this is problematic depends on the final use of the predictive model. In interactive use, this would not be an issue because the user will be able to disambiguate. For actual KBC, the system will have to disambiguate what exact entity it predicted. Here, however, we only check whether the text generated by the model corresponds to one of the gold standard alternatives.

While experimenting, we noticed that in the training and development datasets the names of

entities often correspond with the labels of entities in Wikidata<sup>5</sup>. On Wikidata, these entities also have aliases, and we wanted to know whether we can improve our system by looking up the aliases on Wikidata. This lookup does use language models, so it is *not included as part of the ProP pipeline*, as this would violate the terms of the LM-KBC challenge. Instead, we perform it as an ablation study.

The *alias-fetcher* works as follows. First, we extract a set of types which could be relevant for the specific relation types. For example, country (Q6256) is relevant for *RiverBasinsCountry*. Then, for each relation type, we extract all correctly typed entities, their aliases, and claim count. Then, we take the prediction of the LM, and check whether there is an entity with that label for that relation. If so, we retain the prediction.

Otherwise, we check whether the prediction is equal to any alias. There could be multiple entities for which this is the case. Therefore, we pick the label of the entity with the most claims on Wikidata. The assumption is that it is more likely the answer if it is an ‘interesting’ entity and that these have more claims on Wikidata.

We observe that for four relation types, the changes in the scores are insignificant. For the eight other relations, we see that the F1 score goes up slightly. Overall, this results in an average improvement of the F1 score with 0.014 (Table 3) on the development set. On the test set, we notice a similar improvement. This experiment is not extensive enough to derive definitive conclusions, but it appears to be useful to use structured data to augment the predictions of an LM.

### 5.2.3. Contemporaneity of LMs

We found that questions regarding recent events, particularly those that occurred after 2020, did not yield good predictions by GPT-3 (see 7.3 in the Appendix). This is in line with related findings around LMs and was confirmed by OpenAI<sup>6</sup>. Two examples of this include *Facebook, Inc.* changing its name to *Meta Platforms, Inc.* (in October 2021), and the country of *Swaziland* changing its name to *Eswatini* (in 2018). We also observed similar problems with several instances from the following relations: *PersonProfession*, *PersonCauseOfDeath*, and *PersonPlaceOfDeath*. It is worth noting that it is not important when the model was trained, but whether the training date contains up-to-date information.

## 5.3. Future Work

A natural continuation of this work revolves around improving the individual steps in our pipeline (e.g. fact checking) and their performance, which will directly reflect on the overall macro-F1 score of our approach. Additionally, we could experiment with inverting our pipeline and allow the LM to generate the best prompts by providing the ground truth as input. For example, we could explore techniques that automatically learn prompts, similar to AutoPrompt [15] and OptiPrompt [6], but ideally with a method that requires fewer resources.

In terms of additional components that make use of LMs, we considered developing *meta-prompts* as in [29]. We think it would be interesting to study what meta-prompts can be

---

<sup>5</sup><https://www.wikidata.org>

<sup>6</sup><https://beta.openai.com/docs/guides/embeddings/limitations-risks>

developed for KBC. Is there a set of specific patterns that work better than others? Finally, we could modify our *alias-fetcher* to use the LM to generate well-known aliases for both entities and relations in the training data. This approach could act as a diversification factor, and we believe it will have more freedom in its choice of aliases.

Data augmentation differs from *prompt tuning* in the following ways: While prompt tuning is looking for the optimal prompt to increase the performance for a specific task, *data augmentation* acts as a diversification mechanism for our existing prompts. By employing a more diverse set of prompts we can increase our performance, especially the recall. We base our hypothesis on the fact that knowledge is expressed diversely in the training data (e.g. ambiguity), and we believe this should be considered when prompting an LM.

Furthermore, it would be interesting to further investigate if huge language models are required to perform knowledge graph construction and how to achieve the best prediction performance for the lowest costs.

## 6. Conclusion

We introduced ProP, our "Prompting as Probing" approach to performing knowledge-base completion using a high-capacity pre-trained LM. We showed how we developed different modular components that utilise both LM and the data provided by the organisers to improve ProP's performance, such as the *fact probing* and the *alias fetcher* components. We also investigated well-known techniques around prompt engineering and optimisation and analysed the effect of different prompt formulations on the final performance. However, we conclude that the parameter count of the GPT-3 models is the most significant contributor to performance. Our ProP pipeline outperforms the baseline by 36.4 percentage points on the test split.

Our approach does not only obtain a high macro F1-score on the ground truth but its actual score is suspected to be higher because in several cases where the result was counted as incorrect, the ground truth was either incomplete or used aliases that refer to the same entity as our prediction. Overall, we conclude that language models can be used to augment Knowledge Bases, and we emphasise the difficulty of evaluating question-answering tasks where simple string matching does not suffice.

**Supplemental Material Statement:** Code and data are publicly available from <https://github.com/HEmile/iswc-challenge>.

## Acknowledgements

We thank Frank van Harmelen for his insightful comments. This research was funded by the Vrije Universiteit Amsterdam and the Netherlands Organisation for Scientific Research (NWO) via the *Spinoza* grant (SPI 63-260) awarded to Piek Vossen, the *Hybrid Intelligence Centre* via the *Zwaartekracht* grant (024.004.022), Elsevier's Discovery Lab, and Huawei's DReaMS Lab.

## References

- [1] D. Daza, M. Cochez, P. Groth, Inductive entity representations from text via link prediction, in: Proceedings of the Web Conference 2021, 2021, pp. 798–808.
- [2] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, arXiv preprint arXiv:1908.10063 (2019).
- [3] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, et al., ProtTrans: Towards cracking the language of life’s code through self-supervised deep learning and high performance computing, arXiv preprint arXiv:2007.06225 (2020).
- [4] T. Sorensen, J. Robinson, C. M. Rytting, A. G. Shaw, K. J. Rogers, A. P. Delorey, M. Khalil, N. Fulda, D. Wingate, An information-theoretic approach to prompt engineering without ground truth labels, 2022.
- [5] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language Models as Knowledge Bases?, in: Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2463–2473.
- [6] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, L. Zettlemoyer, Opt: Open pre-trained transformer language models, 2022. arXiv:2205.01068.
- [7] G. Qin, J. Eisner, Learning how to ask: Querying LMs with mixtures of soft prompts, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 5203–5212.
- [8] S. Razniewski, A. Yates, N. Kassner, G. Weikum, Language Models As or For Knowledge Bases (2021). arXiv:2110.04888.
- [9] R. v. Bakel, T. Aleksiev, D. Daza, D. Alivanistos, M. Cochez, Approximate knowledge graph query answering: from ranking to binary classification, in: International Workshop on Graph Structures for Knowledge Representation and Reasoning, Springer, Cham, 2020, pp. 107–124.
- [10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
- [11] H. Li, Language models: past, present, and future, Commun. ACM 65 (2022) 56–63.
- [12] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How Can We Know What Language Models Know?, in: Transactions of the Association for Computational Linguistics 2020 (TACL), volume 8, 2020, pp. 423–438.
- [13] Z. Bouraoui, J. Camacho-Collados, S. Schockaert, Inducing Relational Knowledge from BERT, in: Proc. of the Thirty-Fourth Conference on Artificial Intelligence, AAAI’20, 2020.
- [14] A. Haviv, J. Berant, A. Globerson, BERTese: Learning to speak to BERT, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 3618–3623.
- [15] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts (2020) 4222–4235. arXiv:2010.15980.

- [16] Z. Zhong, D. Friedman, D. Chen, Factual probing is [MASK]: Learning vs. learning to recall (2021) 5017–5033.
- [17] L. Fichtel, J.-C. Kalo, W.-T. Balke, Prompt Tuning or Fine-Tuning - Investigating Relational Knowledge in Pre-Trained Language Models (2021) 1–15.
- [18] T. He, K. Cho, J. Glass, An Empirical Study on Few-shot Knowledge Probing for Pretrained Language Models (2021). [arXiv:2109.02772](#).
- [19] N. Poerner, U. Waltinger, H. Schütze, BERT is Not a Knowledge Base (Yet): Factual Knowledge vs. Name-Based Reasoning in Unsupervised QA 0 (2019). [arXiv:1911.03681](#).
- [20] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, W. W. Cohen, Time-Aware Language Models as Temporal Knowledge Bases (2021). [arXiv:2106.15110](#).
- [21] M. Sung, J. Lee, S. Yi, M. Jeon, S. Kim, J. Kang, Can Language Models be Biomedical Knowledge Bases? (2021) 4723–4734. [arXiv:2109.07154](#).
- [22] Z. Meng, F. Liu, E. Shareghi, Y. Su, C. Collins, N. Collier, Rewire-then-Probe: A Contrastive Recipe for Probing Biomedical Knowledge of Pre-trained Language Models (2021). [arXiv:2110.08173](#).
- [23] J.-C. Kalo, L. Fichtel, P. Ehler, W.-T. Balke, KnowlyBERT - Hybrid Query Answering over Language Models and Knowledge Graphs, in: Proceedings of the International Semantic Web Conference (ISWC), 2020, pp. 294–310.
- [24] H. Arnaout, T.-K. Tran, D. Stepanova, M. H. Gad-Elrab, S. Razniewski, G. Weikum, Utilizing language model probes for knowledge graph repair, in: Wiki Workshop 2022, 2022.
- [25] R. Biswas, R. Sofronova, M. Alam, N. Heist, H. Paulheim, H. Sack, Do Judge an Entity by Its Name! Entity Typing Using Language Models, in: The Semantic Web: ESWC 2021 Satellite Events, 2021, pp. 65–70.
- [26] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, 2019. [arXiv:1909.03193](#).
- [27] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, [arXiv preprint arXiv:1810.04805](#) (2018).
- [28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, [arXiv preprint arXiv:1907.11692](#) (2019).
- [29] L. Reynolds, K. McDonell, Prompt programming for large language models: Beyond the few-shot paradigm, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–7.
- [30] J. Jung, L. Qin, S. Welleck, F. Brahman, C. Bhagavatula, R. L. Bras, Y. Choi, Maieutic prompting: Logically consistent reasoning with recursive explanations, [arXiv preprint arXiv:2205.11822](#) (2022).

## 7. Appendix

### 7.1. Dataset statistics

Here, we provide statistics about the LM-KBC dataset for the training and development split. The statistics of the test split is unknown, because the test split is not public. We assume that the instances from the test are also sampled from a similar data distribution.

**Table 4**

The mean and standard deviation (std) of the number of object-entities per relation. Since some object-entities have alternative labels, we also count the alternative labels. The values are rounded to 2 decimal places.

Relation Type	Number of Object Entities	
	mean	std
CompanyParentOrganization	0.32	0.47
PersonPlaceOfDeath	0.50	0.51
PersonCauseOfDeath	0.52	0.54
PersonLanguage	1.70	1.18
PersonInstrument	1.86	2.81
CountryOfficialLanguage	2.06	2.47
PersonEmployer	2.14	1.65
RiverBasinsCountry	2.28	1.67
ChemicalCompoundElement	3.38	1.12
CountryBordersWithCountry	4.04	2.69
StateSharesBorderState	5.62	2.92
PersonProfession	7.42	4.85

#### 7.1.1. Problems arising from Alternative Labels

The LM-KBC challenge does not include entity linking. Instead, predicted entities are scored against a list of their aliases in the LM-KBC dataset. However, we noticed these lists are often incomplete. For example, for the "National Aeronautics and Space Administration", the extremely common and widely used abbreviation "NASA", is not included in the list of aliases. Another example occurs when the model predicts Aluminum (US and Canadian English) where the ground truth only has Aluminium (British English; a term globally adopted), a lower score gets obtained. Hence, if the model predicts Aluminum or NASA, the predictions are deemed incorrect.

### 7.2. Prompts

Here, we show the templates we used to generate the prompts for the different relations. In our template, we use {subject\_entity} to refer to the head entity for which we are predicting the tail entities for. The generated prompts were used for the following models: Ada, Babbage, Curie and Davinci.

### 7.2.1. CountryBordersWithCountry

Which countries neighbour Dominica?

['Venezuela']

Which countries neighbour North Korea?

['South Korea', 'China', 'Russia']

Which countries neighbour Serbia?

['Montenegro', 'Kosovo', 'Bosnia and Herzegovina', 'Hungary', 'Croatia', 'Bulgaria', 'Macedonia', 'Albania', 'Romania']

Which countries neighbour Fiji?

[]

Which countries neighbour {subject\_entity}?

### 7.2.2. CountryOfficialLanguage

Suriname CountryOfficialLanguage: ['Dutch']

Canada CountryOfficialLanguage: ['English', 'French']

Singapore CountryOfficialLanguage: ['English', 'Malay', 'Mandarin', 'Tamil']

Sri Lanka CountryOfficialLanguage: ['Sinhala', 'Tamil']

{subject\_entity} CountryOfficialLanguage:

### 7.2.3. StateSharesBorderState

San Marino StateSharesBorderState: ['San Leo', 'Acquaviva', 'Borgo Maggiore', 'Chiesanuova', 'Fiorentino']

Whales StateSharesBorderState: ['England']

Liguria StateSharesBorderState: ['Tuscany', 'Auvergne-Rhoone-Alpes', 'Piedmont', 'Emilia-Romagna']

Mecklenberg-Western Pomerania StateSharesBorderState: ['Brandenburg', 'Pomeranian', 'Schleswig-Holstein', 'Lower Saxony']

{subject\_entity} StateSharesBorderState:

#### 7.2.4. RiverBasinsCountry

Drava RiverBasinsCountry: ['Hungary', 'Italy', 'Austria', 'Slovenia', 'Croatia']

Huai river RiverBasinsCountry: ['China']

Paraná river RiverBasinsCountry: ['Bolivia', 'Paraguay', 'Argentina', 'Brazil']

Oise RiverBasinsCountry: ['Belgium', 'France']

{subject\_entity} RiverBasinsCountry:

#### 7.2.5. ChemicalCompoundElement

Water ChemicalCompoundElement: ['Hydrogen', 'Oxygen']

Bismuth subsalicylate ChemicalCompoundElement: ['Bismuth']

Sodium Bicarbonate ChemicalCompoundElement: ['Hydrogen', 'Oxygen', 'Sodium', 'Carbon']

Aspirin ChemicalCompoundElement: ['Oxygen', 'Carbon', 'Hydrogen']

{subject\_entity} ChemicalCompoundElement:

#### 7.2.6. PersonLanguage

Aamir Khan PersonLanguage: ['Hindi', 'English', 'Urdu']

Pharrell Williams PersonLanguage: ['English']

Xabi Alonso PersonLanguage: ['German', 'Basque', 'Spanish', 'English']

Shakira PersonLanguage: ['Catalan', 'English', 'Portuguese', 'Spanish', 'Italian', 'French']

{subject\_entity} PersonLanguage:

#### 7.2.7. PersonProfession

What is Danny DeVito's profession?

['Comedian', 'Film Director', 'Voice Actor', 'Actor', 'Film Producer', 'Film Actor', 'Dub Actor', 'Activist', 'Television Actor']



What is David Guetta's profession?

['DJ']

What is Gary Lineker's profession?

['Commentator', 'Association Football Player', 'Journalist', 'Broadcaster']

What is Gwyneth Paltrow's profession?

['Film Actor', 'Musician']

What is {subject\_entity}'s profession?

### 7.2.8. PersonInstrument

Liam Gallagher PersonInstrument: ['Maraca', 'Guitar']

Jay Park PersonInstrument: ['None']

Axl Rose PersonInstrument: ['Guitar', 'Piano', 'Pander', 'Bass']

Neil Young PersonInstrument: ['Guitar']

{subject\_entity} PersonInstrument:

### 7.2.9. PersonEmployer

Where is or was Susan Wojcicki employed?

['Google']

Where is or was Steve Wozniak employed?

['Apple Inc', 'Hewlett-Packard', 'University of Technology Sydney', 'Atari']

Where is or was Yukio Hatoyama employed?

['Senshu University', 'Tokyo Institute of Technology']

Where is or was Yahtzee Croshaw employed?

['PC Gamer', 'Hyper', 'Escapist']

Where is or was {subject\_entity} employed?

### 7.2.10. PersonPlaceOfDeath

What is the place of death of Barack Obama?

[]

What is the place of death of Ennio Morricone?  
[ 'Rome' ]

What is the place of death of Elon Musk?  
[ ]

What is the place of death of Prince?  
[ 'Chanhassen' ]

What is the place of death of {subject\_entity}?

#### **7.2.11. PersonCauseOfDeath**

André Leon Talley PersonCauseOfDeath: [ 'Infarction' ]

Angela Merkel PersonCauseOfDeath: [ 'None' ]

Bob Saget PersonCauseOfDeath: [ 'Injury', 'Blunt Trauma' ]

Jamal Khashoggi PersonCauseOfDeath: [ 'Murder' ]

{subject\_entity} PersonCauseOfDeath:

#### **7.2.12. CompanyParentOrganization**

Microsoft CompanyParentOrganization: [ 'None' ]

Sony CompanyParentOrganization: [ 'Sony Group' ]

Saab CompanyParentOrganization: [ 'Saab Group', 'Saab-Scania',  
'Spyker N.V.', 'National Electric Vehicle Sweden'', 'General Motors' ]

Max Motors CompanyParentOrganization: [ 'None' ]

{subject\_entity} CompanyParentOrganization:

### **7.3. Failure cases**

Here, we list three failure examples for each relation for the Davinci model. A comprehensive list of failure cases can be found under [https://github.com/HEmile/iswc-challenge/tree/main/failure\\_cases](https://github.com/HEmile/iswc-challenge/tree/main/failure_cases).

#### **7.3.1. CountryBordersWithCountry**

SubjectEntity: Bahrain

Ground Truth: ['iran', 'saudi arabia']  
GPT-3 Prediction: ['qatar', 'saudi arabia', 'united arab emirates']

SubjectEntity: Barbados  
Ground Truth: []  
GPT-3 Prediction: ['trinidad and tobago']

SubjectEntity: Cuba  
Ground Truth: ['united states of america', 'usa']  
GPT-3 Prediction: ['bahamas', 'haiti', 'jamaica', 'turks and caicos islands', 'united states']

### 7.3.2. CountryOfficialLanguage

SubjectEntity: Afghanistan  
Ground Truth: ['arabic', 'baluchi', 'dari', 'nuristani', 'pamir', 'pashayi', 'pashto', 'turkmen', 'uzbek']  
GPT-3 Prediction: ['dari', 'pashto']

SubjectEntity: Botswana  
Ground Truth: ['english']  
GPT-3 Prediction: ['setswana']

SubjectEntity: Zimbabwe  
Ground Truth: ['barwe', 'chewa', 'english', 'kalanga', 'khoisan', 'nambya', 'ndau', 'ndebele', 'northern ndebele', 'sesotho', 'shona', 'tonga', 'tsonga', 'tswana', 'venda', 'xhosa']  
GPT-3 Prediction: ['chewa', 'english', 'ndebele', 'shangaan', 'shona', 'sotho', 'tonga', 'venda']

### 7.3.3. StateSharesBorderState

SubjectEntity: Andalusia  
Ground Truth: ['beja', 'castile-la mancha', 'extremadura', 'faro', 'gibraltar', 'murcia', 'region of murcia']  
GPT-3 Prediction: ['castilla-la mancha', 'ceuta', 'extremadura', 'melilla', 'murcia']

SubjectEntity: Obwalden  
Ground Truth: ['canton of bern', 'canton of lucerne', 'lucerne', 'nidwalden', 'schwyz', 'uri']  
GPT-3 Prediction: ['berne', 'uri', 'valais']

SubjectEntity: Zagreb

Ground Truth: ['krapina-zagorje', 'krapina-zagorje county',  
'zagreb', 'zagreb county']  
GPT-3 Prediction: ['bosnia and herzegovina', 'serbia', 'slovenia']

#### 7.3.4. RiverBasinsCountry

SubjectEntity: Didi Liakhvi river  
Ground Truth: ['georgia']  
GPT-3 Prediction: ['georgia', 'russia']

SubjectEntity: Dniester  
Ground Truth: ['moldova', 'ukraine']  
GPT-3 Prediction: ['moldova', 'romania', 'ukraine']

SubjectEntity: San  
Ground Truth: ['poland', 'ukraine']  
GPT-3 Prediction: ['mexico']

#### 7.3.5. ChemicalCompoundElement

SubjectEntity: Aluminum Iodide  
Ground Truth: ['aluminium', 'iodine']  
GPT-3 Prediction: ['aluminum', 'iodine']

SubjectEntity: Atracurium Besylate  
Ground Truth: ['carbon', 'hydrogen', 'nitrogen', 'oxygen', 'sulfur']  
GPT-3 Prediction: ['atracurium', 'hydrogen', 'oxygen']

SubjectEntity: Flavin Adenine Dinucleotide  
Ground Truth: ['carbon', 'hydrogen', 'nitrogen', 'oxygen', 'phosphorus']  
GPT-3 Prediction: ['adenine', 'dinucleotide', 'flavin']

#### 7.3.6. PersonLanguage

SubjectEntity: A. R. Rahman  
Ground Truth: ['hindi', 'tamil', 'telugu']  
GPT-3 Prediction: ['bengali', 'hindi', 'kannada', 'malayalam',  
'marathi', 'tamil', 'telugu']

SubjectEntity: Akon  
Ground Truth: ['english']  
GPT-3 Prediction: ['english', 'french', 'wolof']

SubjectEntity: Genelia D'Souza  
Ground Truth: ['hindi', 'malayalam', 'tamil', 'telugu']

GPT-3 Prediction: ['bengali', 'english', 'hindi', 'marathi', 'telugu']

### 7.3.7. PersonProfession

SubjectEntity: Elon Musk

Ground Truth: ['engineer', 'entrepreneur', 'inventor', 'programmer']

GPT-3 Prediction: ['businessperson', 'engineer', 'inventor', 'investor', 'philanthropist', 'product designer']

SubjectEntity: J. K. Rowling

Ground Truth: ['author', "children's writer", 'executive producer', 'film producer', 'novelist', 'producer', 'producer', 'screenwriter', 'writer', 'writer']

GPT-3 Prediction: ['author', 'film producer', 'philanthropist', 'screenwriter']

SubjectEntity: Lady Gaga

Ground Truth: ['singer', 'songwriter']

GPT-3 Prediction: ['actress', 'dancer', 'philanthropist', 'record producer', 'singer', 'songwriter']

### 7.3.8. PersonInstrument

SubjectEntity: Ronan Keating

Ground Truth: []

GPT-3 Prediction: ['piano']

SubjectEntity: Taeyang

Ground Truth: []

GPT-3 Prediction: ['guitar']

SubjectEntity: Chester Bennington

Ground Truth: []

GPT-3 Prediction: ['guitar', 'piano']

### 7.3.9. PersonEmployer

SubjectEntity: Kent Beck

Ground Truth: ['meta platforms']

GPT-3 Prediction: ['facebook', 'three rivers institute']

SubjectEntity: Serena Williams

Ground Truth: ['unicef']

GPT-3 Prediction: ['tennis']

SubjectEntity: Guido van Rossum  
Ground Truth: ['microsoft']  
GPT-3 Prediction: ['dropbox', 'google']

#### **7.3.10. PersonPlaceOfDeath**

SubjectEntity: Avicii  
Ground Truth: ['muscat']  
GPT-3 Prediction: ['muscat, oman']

SubjectEntity: John Coltrane  
Ground Truth: ['huntington']  
GPT-3 Prediction: ['new york city']

SubjectEntity: Rachel Caine  
Ground Truth: ['texas']  
GPT-3 Prediction: ['']

#### **7.3.11. PersonCauseOfDeath**

SubjectEntity: Ahmed Zewail  
Ground Truth: ['lymphoma', 'spinal cord lymphoma']  
GPT-3 Prediction: ['cancer']

SubjectEntity: Avicii  
Ground Truth: ['exsanguination']  
GPT-3 Prediction: ['suicide']

SubjectEntity: Ennio Morricone  
Ground Truth: ['femoral fracture', 'fracture']  
GPT-3 Prediction: ['']

#### **7.3.12. CompanyParentOrganization**

SubjectEntity: Aston Martin lagonda  
Ground Truth: []  
GPT-3 Prediction: ['aston martin lagonda global holdings plc']

SubjectEntity: Austro-Daimler  
Ground Truth: []  
GPT-3 Prediction: ['daimler ag']

SubjectEntity: Hyundai Motor Company  
Ground Truth: ['hyundai']  
GPT-3 Prediction: ['hyundai motor group']

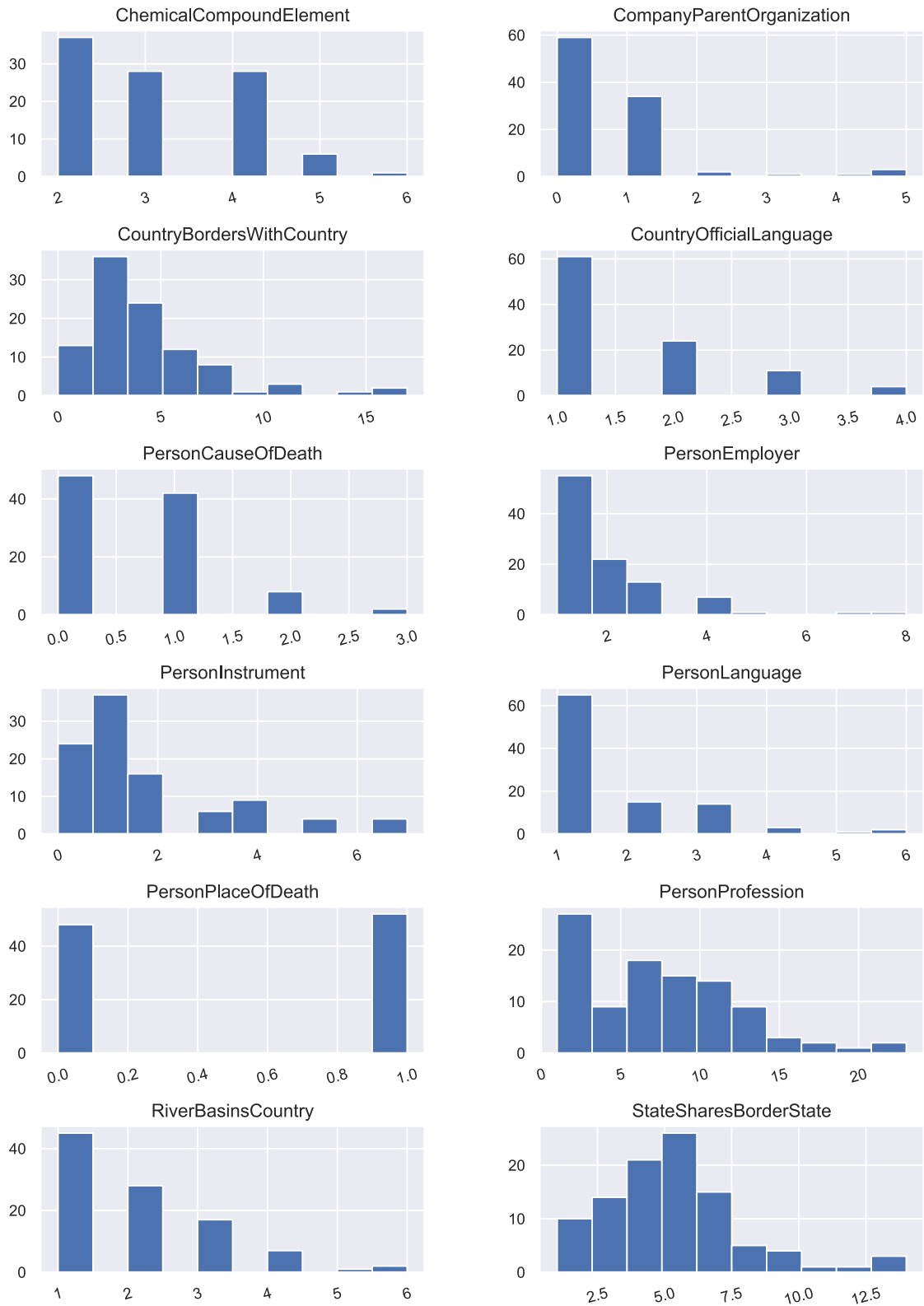
## 7.4. Language Model Size

Table 5 shows the values of the scaling experiments. These values were used to produce Figure 1.

**Table 5**

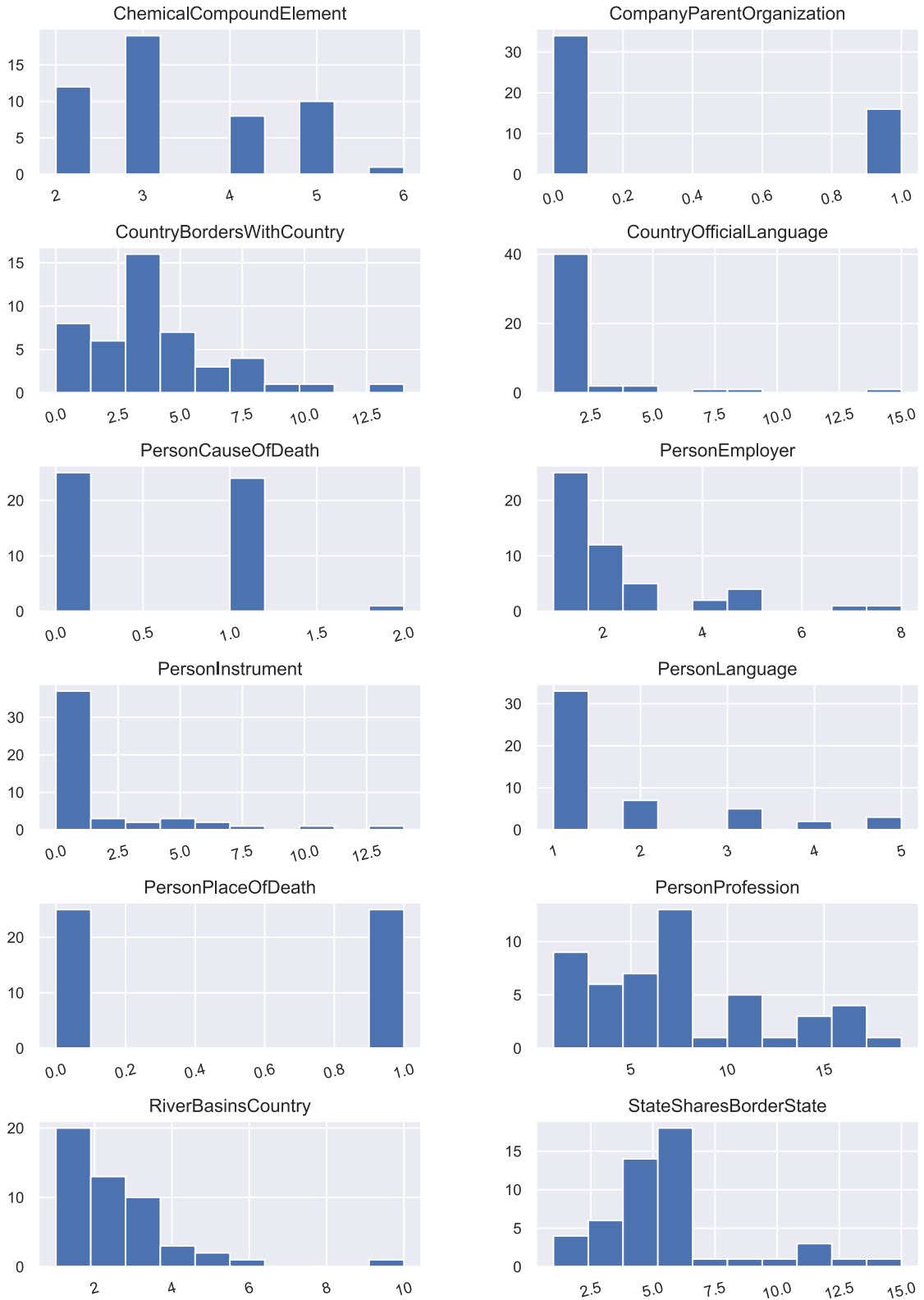
Precision, Recall and F1-score for the GPT-3 models with varying embedding dimensions. Best scores are in **bold**.

Method	Precision	Recall	F1-score
Baseline (BERT)	0.175	0.129	0.140
Ada	0.180	0.194	0.161
Babbage	0.325	0.263	0.269
Curie	0.378	0.375	0.343
Davinci	<b>0.707</b>	<b>0.694</b>	<b>0.677</b>



**Figure 2:** The number of answers per relation type for the training set provided by the organisers.





**Figure 3:** The number of answers per relation type for the development set provided by the organisers.