LANGUAGE MODELS ARE OPEN KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

The paper presents knowledge graphs (KGs) that are constructed from pre-trained language models (e.g., BERT, GPT-2) without human supervision. Popular KGs (e.g, Wikidata, NELL) are built in either a supervised or semi-supervised manner, requiring human to manually create the knowledge. Recent deep language models automatically acquire human knowledge from large-scale corpora via pre-training. The embedded knowledge has enabled the language models to perform NLP tasks at the human level, e.g., answering questions, writing codes and articles. In this paper, we propose an unsupervised method to recall the learned knowledge in language models as KGs. We find the quality of the generated KGs promising by comparing with two KGs (Wikidata, TAC KBP) created by human. Surprisingly, our KGs provide a reasonable amount of open factual knowledge that is new and not in the reference KGs. The code and our KGs are publicly available upon publication.

1 INTRODUCTION

Knowledge graph (KG) is an important resource for machines to acquire intelligence. Factual knowledge in KG is injected into AI applications to mimic important skills possessed by human, e.g., reasoning and understanding. KG construction is mainly supervised, requiring human to hand-write every fact, such as Freebase (Bollacker et al., 2008) and Wikidata. KG can also be constructed in a semi-supervised way, in which a semi-automatic extractor is used to obtain the facts from web corpus (e.g., NELL (Carlson et al., 2010) and Knowledge Vault (Dong et al., 2014)). Human however still needs to interact with the extractor to improve the quality of the discovered facts. Therefore, human knowledge is required in constructing KGs.

Recent progress in language models (LMs), e.g., BERT (Devlin et al., 2018) and GPT-2/3 (Radford et al., 2019; Brown et al., 2020), has led to superior results even outperforming human in a wide range of tasks, e.g., sentence classification (Wang et al., 2018), question answering (Brown et al., 2020). Surprisingly, pre-trained LMs are capable to write poetry, music, and code, while such tasks often require we human to spend a reasonable amount of time in learning the relevant knowledge to work well. In fact, pre-trained LMs automatically acquire human knowledge from large-scale corpora (e.g., BookCorpus (Zhu et al., 2015), WebText2 (Brown et al., 2020)) via pre-training. The learned knowledge in pre-trained LMs is the key to the current success. We therefore ask the following question: instead of using the manually created human knowledge, *can we directly use the human knowledge embedded in pre-trained LMs to construct KGs?*

In this paper, we design an unsupervised approach MAMA that successfully recalls the factual knowledge in LMs to build KGs from scratch. As illustrated in Figure 1, MAMA has two stages: Match and Map. **Match** stage generates a set of open facts from the target corpus via matching the knowledge in pre-trained models. General or world knowledge from large-scale corpora is embedded in LMs, thus open facts in the target corpus are covered by the LMs. The open facts are matched with the knowledge in LMs through beam search, which walks through the attention weight matrices of the pre-trained LMs to maintain the k-best matched candidate facts. For example, an open fact (Dylan, is, songwriter) from the sentence "Dylan is a songwriter." is returned by the search. **Map** stage produces an open KG via mapping the open facts to both fixed KG schema and open schema. Although the well-matched open facts are with reasonable quality, we recognize duplicated facts as a major issue due to the diverse nature of language expressions. For example, we have two identical

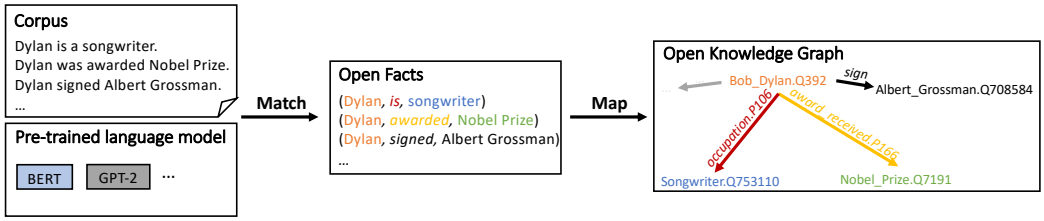


Figure 1: Overview of the proposed approach MAMA. Given the input: a textual corpus containing passages and sentences, e.g., English Wikipedia, and a pre-trained language model (LM), e.g., BERT, GPT-2/3, MAMA first generates a set of open facts from the target corpus via *matching* the knowledge in pre-trained LMs, then produces an open knowledge graph by *mapping* the matched open facts to both KG schema (e.g., *(Bob_Dylan.Q392, occupation.P106, Songwriter.Q753110)*) and open schema (e.g., *(Bob_Dylan.Q392, sign, Albert_Grossman.Q708584)*).

open facts (Dylan, is, songwriter) and (Bob, is, songwriter). We deduplicate the facts by grounding them to a well defined KG schema, which results in a single KG fact (Bob_Dylan.Q392, occupation.P106, Songwriter.Q753110). The incompleteness is a longstanding issue of KGs due to the fixed schema, impeding the wider adoption in downstream applications. We therefore combine the partially or totally ungrounded facts with the grounded facts. This results in a new type of KG, *open KG*, with a mixture of fixed KG schema and open schema.

Our contributions are as follows:

1. We study *how to construct KGs from scratch using pre-trained LMs*, which can be beneficial to relevant areas, such as neural model explanation and information extraction.
2. We propose an unsupervised two-stage approach MAMA to first *match* the open facts with knowledge in LMs, then *map* the open facts to produce a KG.
3. We generate a new type of KG, namely open KG, consists of *mapped facts* with promising quality by comparing with two oracle KGs (Wikidata and TAC KBP); and *unmapped facts* for KG completeness.

2 MAMA

We introduce an unsupervised end-to-end approach Match-Map (MAMA) as illustrated in Figure 1 to construct an open knowledge graph from language models. The two stages of MAMA are:

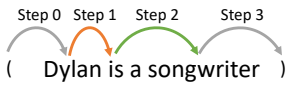
Match generates a set of open facts from a target corpus. LMs possess global or world knowledge learned from large-scale corpora, which doesn't perfectly match the target corpus. The goal of this stage is to match the knowledge in pre-trained LMs with open facts in the corpus. Each fact is represented as (head, relation, tail) [1], in short, (h, r, t), and passed to Map stage. Match procedure is detailed in Sec. 2.1.

Map produces an open knowledge graph (KG). Open facts that matched well with LMs are of good quality. However, open facts suffer from duplication issues due to the diversity of the language expression, e.g., "Dylan" and "Bob" could both refer to "Bob Dylan" or someone else. Mapping open facts to the schema of an existing KG could relieve the issue. A longstanding issue of traditional KGs is the incompleteness caused by the fixed KG schema, impeding the wider adoption of KGs in downstream tasks. We therefore construct an open KG that has two portions: (1) open facts that are grounded to the given schema, e.g., (Dylan, is, songwriter) is mapped to (Bob_Dylan.Q392, occupation.P106, Songwriter.Q753110) according to Wikidata schema; and (2) open facts that are either partially grounded or cannot be grounded, e.g., (Dylan, signed, Albert Grossman). This stage is described in Sec. 2.2.

2.1 MATCH

We frame the matching procedure as a search problem. Given a sentence, the open facts are matched with the knowledge in LMs via searching through the attention weight matrices of the pre-trained

[1] We use the term "head" and "tail" to denote head and tail's "entity" or "entity mention" for simplicity.



Step	Action	Intermediate candidates	Matching degrees
0	start	(Dylan,	0
1	yield	(Dylan, is	0.3
2	yield	(Dylan, is songwriter	0.7
3	stop	(Dylan, is, songwriter)	0.7

(a) Matching example.

Key: \ Query:	Dylan	is	a	songwriter
Dylan	x	x	x	x
is	0.3	x	x	x
a	0.1	0.2	x	x
songwriter	0.1	0.4	0.2	x

(b) Matching degree calculation in attention matrix.

Figure 2: Illustration of Match stage. The upper part of (a) represents the general matching steps of generating an open fact (Dylan, is, songwriter) from the sentence "Dylan is a songwriter." The lower portion shows the corresponding step-by-step process. At each step, the search chooses one of the actions, i.e., *start, yield, stop* to produce an intermediate candidate according to the matching degree. The matching degree is added up by the next largest attention weight from attention matrix (b), e.g., "is" (0.3 in the orange square) is yielded as the next token of "Dylan" in step 1 (in the orange arrow). "x" in (b) marks the positions to prevent searching backward. The facts with the top matching degrees are returned.

LMs to maintain best matched candidate facts. The matching degree is derived from the sentence's associated self-attention weight matrix, which is the main container of the knowledge information in pre-trained LMs. Another advantage of using self-attention is that the relational connections or dependencies within a sentence have been pre-built by considering the sentence as both query and key (Vaswani et al., 2017). Intuitively, the larger the attention weight between two positions/subwords is, the higher the matching degree is.

2.1.1 Beam Search

We design a simple-yet-effective beam search to capture the intuition. For every head-tail pair (h, t) in a sentence from the corpus, the search maintains the k-best facts of the pair. The matching degree of each candidate fact is scored by aggregating the attention weights in the pre-trained LM. Let's first consider search from left to right with beam size equals to 1. An example search process is shown in Figure 2. Given a head-tail pair (Dylan, songwriter), at each step, the search performs one of the following actions:

Start the search by adding the head as a candidate into the beam. For simplicity, we use $\mathbf{start}(h)$ to denote the action, which returns $(h,)$. In Figure 2(a), at step 0, "Dylan" is added as the head of the intermediate candidate. The matching degree is initialized to 0.

Yield a new intermediate candidate by appending the next mostly attended position to the current candidate. At step 1 (orange arrow in Figure 2(a)), "is" is selected since it has the largest attention weight connecting to the last position in the candidate "Dylan", i.e., the attention score is 0.3 as highlighted in orange as shown in Figure 2(b). The multi-head attention is reduced to a single head so that every two positions in the input sentence are associated with one attention weight. We experiment with different reduction setups in Sec. 3. "x" marks the positions that are not considered in the search, i.e., positions prior to the current position, to prevent searching backward. Step 2 similarly takes $\mathbf{yield}$ action, the matching degree adds up to 0.7 with the largest attention weight 0.4 (in green) between "songwriter" and "is". We use $\mathbf{yield}(c, \mathbf{A}, s)$ to represent the above procedure, where c is a candidate, $\mathbf{A}$ is the attention matrix, and s represents the sentence, which returns a new candidate.

Stop the search step if reaching the tail, and produce a valid open fact. As beam size equals to 1, (Dylan, is, songwriter) is the only returned open fact for the given pair. We denote this step using $\mathbf{stop}(c, t)$, which returns a valid fact.

Algorithm 1 Beam search for matching facts.

Input: Head-tail pair (h, t), sentence s, attention matrix $\mathbf{A}$ of s, action manager $\mathcal{O} = \{\mathbf{start}, \mathbf{yield}, \mathbf{stop}\}$, beam size k
Output: Open facts $\mathbb{T}_{(h,t)}$

1: $\mathbb{T}_{(h,t)} \leftarrow \{\mathbf{start}(h)\}$
2: **while** $\exists c \in \mathbb{T}_h[\mathcal{O}(c) = \mathbf{yield}]$ **do**
3: **for each** $c \in \mathbb{T}_{(h,t)}$ **do**
4: $\widetilde{\mathbb{T}}_{(h,t)} \leftarrow \emptyset$
5: **if** $\mathcal{O}(c) = \mathbf{yield}$ **then**
6: $\widetilde{\mathbb{T}}_{(h,t)} \leftarrow \widetilde{\mathbb{T}}_{(h,t)} \cup \{\mathbf{yield}(c, \mathbf{A}, s)\}$
7: **else**
8: $\widetilde{\mathbb{T}}_{(h,t)} \leftarrow \widetilde{\mathbb{T}}_{(h,t)} \cup \{\mathbf{stop}(c, t)\}$
9: **end if**
10: **end for**
11: $\mathbb{T}_{(h,t)} \leftarrow \mathrm{TOP}(k, \widetilde{\mathbb{T}}_h)$
12: **end while**
13: **return** $\mathbb{T}_{(h,t)}$

The details of the proposed beam search are in Algorithm 1. We preprocess textual corpora by using noun chunk [2] to identify the potential head and tail. Once the search is started (line 1), while there are still new candidates waiting to be yield (line 2), the search continues, and the top k (beam size) candidates sorted by the matching degrees are reserved (line 3-11). In practice, we implement an action manager $\mathcal{O}$ to decide which action to take at each step. We also notice some facts are in reverse order in the sentence, thus enable bidirectionality by running the algorithm in both directions (left to right and right to left). The beam search is implemented by the breadth-first search, which is efficient as the time complexity is $O(k \cdot d)$, where d is the maximum depth of the search tree.

2.1.2 FILTER

Although the basic functionality provided by beam search is sufficient for finding useful open facts, we have found a few constraints useful. Given a fact (h, r, t) from beam search result $\mathbb{T}_{(h,t)}$, it remains as a fact if satisfying all the following constraints.

The matching degree of (h, r, t) is above a threshold. We compare the matching degrees corpus-wide to only reserve the facts that are matched better with the knowledge in LMs in the corpus. For example, MAMA extracts a fact (Rolling Stone, wrote, pop song) from *Rolling Stone wrote: "No other pop song has so thoroughly challenged artistic conventions"*, which is not an accurate fact given the sentence. We observe the associated matching degree is below the threshold, while the matching degrees of high-quality facts from the same documents, e.g., (Dylan, is, songwriter), or confident facts from the other documents are beyond the threshold.

Relation r is a contiguous sequence in the sentence. We can avoid r that has no meaningful interpretation (Fader et al., 2011), e.g., (Rolling Stone, wrote challenged, conventions) from the above sentence.

The distinct frequency of r is above a threshold. To avoid facts to be over-specified, e.g., (Dylan, signed to Sam Peckinpah's film, Pat Garrett and Billy the Kid), we require r should take many distinct head-tail pairs in the corpus.

2.2 MAP

The objective of Map stage is to generate an open KG. Map first tries to ground the open facts from Match stage to the schema of an existing KG, e.g., Wikidata, to generate an intermediate KG, then adds the partially grounded facts and unmapped open facts to the intermediate KG to eventually produce the open KG. The benefits brought by this stage are two-fold: (1) *Deduplicate the open facts*: if two entities refer to the same KG entry, e.g., "Bob" and "Dylan", they are both mapped to

[2] https://spacy.io/usage/linguistic-features/#noun-chunks

a single entity in KG, e.g., "Bob_Dylan.Q39" in Wikidata. Relations are similarly deduplicated. (2) *Enable the evaluation between open facts and KG*: we are able to quantitatively measure the quality of the constructed KG by comparing the grounded facts with the oracle KG.

2.2.1 GROUND TO KG SCHEMA

The goal is to ground an open fact (h, r, t) to a fact (h_k, r_k, t_k) in an existing KG. The reason for grounding to an existing schema is to make use of the high-quality schema designed by experts (to avoid duplicated efforts of building from scratch) and enables to evaluate the open facts with KG facts contributed by humans. We first ground both entities h, t to h_k, t_k, then ground the relation phrase r to r_k in KG.

Entity linking to KG. We leverage an unsupervised entity linker based on a mention-to-entity dictionary (Spitkovsky & Chang, 2012) to link the entities for scalability consideration. Besides, contextual information is crucial to link the entities correctly, we further use the word embedding of the context to disambiguate the entities.

Relation mapping with KG. We largely follow the relation mapping method proposed by Angeli et al. (2015) to construct an offline relation map for relation grounding purposes. The basic idea is that the more often linked entity pairs (i.e., entities are with type information from the entity linking step) co-occur between the open facts and KG facts, the more likely the corresponding relations are mapped to each other. In addition, we normalize each relation phrase by lemmatization, and removing inflection, auxiliary verbs, adjectives, adverbs. One author manually annotates the top 20 relations mapped to each KG relation in approximately one day, and preserves the verified relation mapping.

2.2.2 MAP TO OPEN SCHEMA

There are two types of ungrounded open facts. First, open facts are partially grounded. An example is (Dylan, signed, Albert Grossman) in Figure 1, where both head and tail are grounded to Wikidata, but the relation is not mapped. Second, open facts are completely not grounded, where all head and tail, and relation stay open. We simply add both cases to the constructed KG from Sec. 2.2.1 to produce an open KG.

The resulting Open KG is a new type of KG that mixes the fixed schema of an existing KG with flexible open schema, suggesting new directions for the next generation of KGs and benefit the downstream KG based applications, e.g., QA and commonsense reasoning. Moreover, the process of producing an open KG provides a principled way from a semi-structured triplet set to a structured graph. The structured graph can be easily visualized and analyzed to further benefit the community.

3 EXPERIMENTS

How well can language models generate knowledge graphs? We experimentally explore how well can MAMA answer the question in the section. To measure the ability of LMs in generating KGs, we directly measure the quality of resulting open KGs. The open KG contains two types of facts: mapped facts that are grounded to a fixed schema of KG; and ungrounded open facts. We first quantitatively evaluate MAMA by comparing the mapped facts to oracle KGs annotated by humans in Sec. 3.1, and analyze the unmapped facts in-depth in Sec. 3.2.

3.1 RESULTS ON MAPPED FACTS

We first study the quality of the grounded facts. As the facts have been mapped to the oracle KGs, we are able to quantitively compare the open facts with the oracle facts in the reference KGs.

3.1.1 DATASETS

We compare the mapped facts from MAMA with the facts in two KGs:

TAC KBP. TAC Knowledge Base Population (KBP) Slot Filling is a task to search a document collection to fill in the tail/object entity for predefined relations (slots) for a given head/subject entity in a reference KG. We experiment with the reference KG in the 2013 challenge. We use the document collection of the 2013 task. The statistic of the dataset is shown in Table 1.

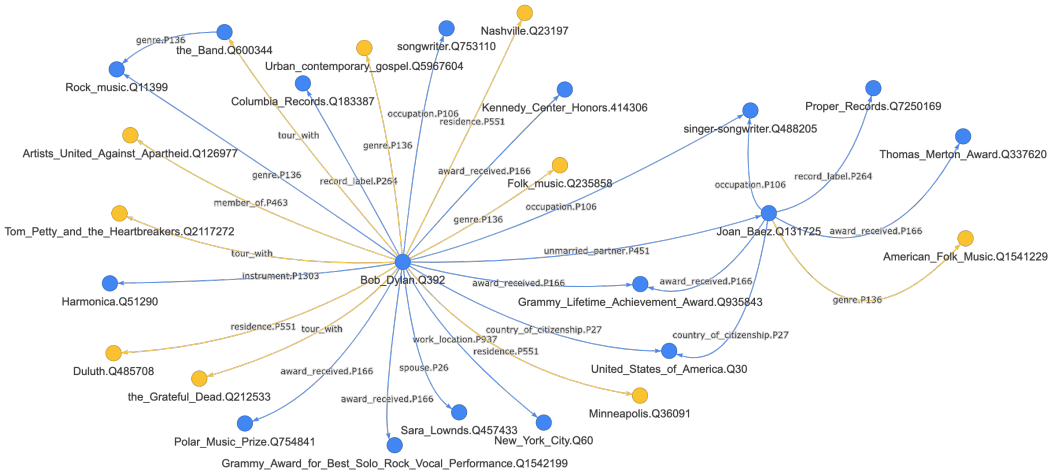


Figure 3: An example open KG generated by MAMA using $\text{BERT}_{\text{BASE}}$ from Wikipedia page entitled "Bob Dylan". The blue arrow represents the mapped facts to Wikidata schema, while the yellow arrow denotes the unmapped facts with open schema.

KG	# of facts	# of documents
TAC KBP	1,458	3,877,207
Wikidata	13,124,592	6,047,494

Table 1: Dataset statistics of two knowledge graphs: TAC KBP and Wikidata. TAC KBP refers to TAC KBP Slot Filling 2013 challenge. # of documents for TAC KBP is the number of the documents in the 2013 task. # of documents for Wikidata is the size of English Wikipedia.

Wikidata. We use popular Wikidata as another KG. We use the English Wikipedia pages as the text corpus, since a reasonable amount of facts in Wikidata is from Wikipedia. The statistic is in Table 1.

3.1.2 METHOD DETAILS

To evaluate the mapped facts, we first use Match stage of MAMA to run over the corresponding documents to generate the open facts. For Map stage on TAC KBP, we use the oracle annotation of the entities in the TAC KBP corpus. On Wikidata, the entity linking method described in Sec. 2.2.1 is leveraged to link entities in the triplets to Wikipedia anchors, and a Wikipedia anchor to Wikidata item dictionary is used to further link the entities to Wikidata. The relation mapping is constructed offline for TAC KBP and Wikidata respectively using the method in Sec. 2.2.1.

The parameter settings are shared on TAC KBP and Wikidata. The beam size of Algorithm 1 is set to 6 according to parameter study in Figure 4(a). For the filtering constraints described in Sec. 2.1.2, the matching degree threshold is set to 0.003 according to the parameter study (Figure 4(b)). The number of distinct entity pairs is set following (Fader et al., 2011). To generate the attention weight matrix $\mathbf{A}$ of a sentence, we reduce the weights of every attention head in the last layer of pre-trained LMs using mean operator based on the parameter study in Figure 4(c). We set the max sequence length to 256, and batch size as 32. We use pre-trained LM implementations in Transformers package [3]. We use sentence splitter [4] to segment the documents into sentences.

3.1.3 TAC KBP

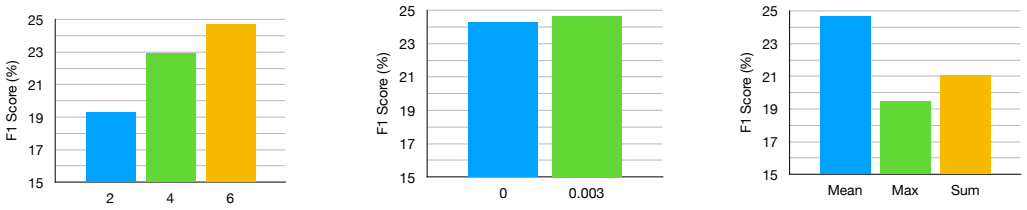
To verify the ability to produce correct facts, we compare open facts from MAMA to the outputs of an open information system, which also produces open triplets in the form of (h, r, t).

Stanford OpenIE leverages POS tag and dependency parser, and generates self-contained clauses from longer sentences to extract the triplets, which is the best open information extraction sys-

[3] https://github.com/huggingface/transformers
[4] https://spacy.io/api/sentencizer

Method	Precision%	Recall%	F1%
Stanford OpenIE (Angeli et al., 2015)	55.18	14.61	23.10
MAMA-GPT-2 (ours)	61.54	14.68	23.70
MAMA-GPT-2$_{\text{MEDIUM}}$ (ours)	62.02	15.56	24.87
MAMA-BERT$_{\text{BASE}}$ (ours)	61.48	15.43	24.67
MAMA-BERT$_{\text{LARGE}}$ (ours)	61.61	16.00	**25.40**

Table 2: Compare the quality of grounded facts on TAC KBP.



(a) Beam size. (b) Matching degree threshold. (c) Attention weights reduction.

Figure 4: Parameter study with MAMA-BERT$_{\text{BASE}}$ on TAC KBP.

tem (Angeli et al., 2015) on TAC KBP Slot Filling 2013 (Surdeanu, 2013). After collecting the triplets from the system, we use the same mapping procedure as in Sec. 2.2 to ground the triplets to the corresponding KG.

We use two families of pre-trained LMs with MAMA. First, We first use BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$ with MAMA, namely **MAMA-BERT**$_{\text{BASE}}$ and **MAMA-BERT**$_{\text{LARGE}}$. Second, GPT-2s are used, i.e., **MAMA-GPT-2**, **MAMA-GPT-2**$_{\text{MEDIUM}}$, that have similar sizes compared to two BERT models.

Table 2 shows the results on TAC KBP. We use the official TAC KBP Slot Filling 2013 scorer to evaluate precision, recall and F1 on TAC KBP. From the results, we find that MAMA-BERT$_{\text{LARGE}}$ performs the best. MAMA-BERT$_{\text{LARGE}}$ outperforms the previous state-of-the-art Stanford OpenIE by over 6% in F1 [5]. This shows the proposed end-to-end MAMA is able to construct relatively high-quality KG only based on BERT without relying on any linguistic features, such as POS tag and dependency parser. Besides, we find GPT-2s generally work well, and larger and deep LMs potentially lead to better quality. This complies with our intuition on more knowledge is embedded in deeper models. We could further improve the results if larger pre-trained models are available, e.g., GPT-3.

3.1.4 WIKIDATA

We select the best method MAMA-BERT$_{\text{LARGE}}$ to compare with Stanford OpenIE for scalability experiments on Wikidata. We follow the same definition as the slot filling task to calculate precision, recall and F1 on Wikidata. Table 3 summarizes the results. MAMA-BERT$_{\text{LARGE}}$ outperforms Stanford OpenIE approximately 6%. The results again show the effectiveness of MAMA in generating open facts and benefiting KG construction. The results also indicate that MAMA is scalable to large KGs. The finding is particularly important, since we are now able to construct larger KGs from scratch when larger datasets are used.

3.2 CASE STUDY ON UNMAPPED FACTS

We turn to study the quality of the ungrounded facts in the open KG generated by MAMA. We manually check the unmapped facts and observe a promising trend. We find a reasonable percentage of open facts are correct but are not facts in the KG. We find 17.7% of such ungrounded open facts on TAC KBP, and 41.4% on Wikidata. The reason is that the open fact is supported by the reference KG's schema, but the instance is not in the KG, e.g., (Bob_Dylan.Q392, residence.P551, Nashville.Q23197) as shown in Figure 3. Another reason is that the open fact is not in the KG schema

[5] We have worked with the authors of Stanford OpenIE to reproduce the numbers in Angeli et al. (2015).

Method	Precision%	Recall%	F1%
Stanford OpenIE (Angeli et al., 2015)	21.48	13.02	16.21
MAMA-BERT$_{\text{LARGE}}$ (ours)	28.71	13.38	**18.26**

Table 3: Compare the quality of grounded facts on Wikidata.

at all, e.g., (Bob_Dylan.Q392, tour_with, the_Grateful_Dead.Q212533). The unmapped facts could help improve the completeness of the KG. As illustrated in Figure 3, the open KG is a new type of KG combining fixed schema with open schema. The improved coverage could be beneficial to downstream applications, such as QA and commonsense reasoning.

3.3 Error Analysis

Although MAMA achieves reasonable evaluation scores, there is still significant room to improve. To further understand the shortage of MAMA, we conduct an error analysis of the incorrect facts. On Wikidata, 27.4% of the errors are caused by incorrect entities, while the relation phrases are correct. The errors are due to the incorrect noun chunk detected by the spaCy. 24.1% of the errors are due to the relation mapping created in Sec. 2.2.1. The rest errors made by MAMA are incorrect relation phrases, such as uninformative relation phrases, e.g., (Dylan, made, his breakthrough), which is similar to the errors made by open IE systems (Fader et al., 2011), which again shows the linguistic knowledge is in pre-trained LMs. We find similar errors are made by MAMA on TAC KBP. We plan to identify entities directly using LMs, and learn a more robust relation mapping.

4 Related Work

Knowledge graph construction can be generally categorized into two groups, 1) supervised approaches. Wikidata, Freebase (Bollacker et al., 2008), are built based on human supervision (Suchanek et al., 2007; Hoffart et al., 2013; Auer et al., 2007); 2) semi-supervised approaches. Open information extraction systems (Fader et al., 2011; Schmitz et al., 2012; Angeli et al., 2015) leverage linguistic features to extract triplets for open schema KG. Besides, semi-automatic systems (Carlson et al., 2010; Niu et al., 2012; Dong et al., 2014) extract information based on a fixed schema or ontology. MAMA instead uses learned knowledge in pre-trained LMs to unsupervisedly construct an open KG, which is a mixture of fixed schema and open schema.

Language models, e.g., BERT (Devlin et al., 2018), GPT-2/3 (Radford et al., 2019; Brown et al., 2020), ELMo (Peters et al., 2018) contain factual knowledge obtained via pre-training on large-scale corpora such as Wikipedia and BookCorpus (Zhu et al., 2015). Studies have leveraged the pre-trained LMs as virtual KGs, and show reasonable performance in QA tasks (Dhingra et al., 2020; Guu et al., 2020) and language modeling (Khandelwal et al., 2019). LMs are further enhanced by KGs (Zhang et al., 2019; Peters et al., 2019) to show improvements in knowledge-driven tasks. While the existing work utilizes knowledge in an implicit way, the main difference is that our approach explicitly extracts facts from the LMs.

BERTology generally refers to understanding or explaining deep neural networks. There has been lot of work to understand what the neural networks learn (Linzen et al., 2016; Adi et al., 2016; Tenney et al., 2019). With regard to analyzing BERT, substantial recent work focuses on both visualizing and analyzing the attention (Vig, 2019; Jain & Wallace, 2019; Michel et al., 2019; Clark et al., 2019). Instead of analyzing or visualizing, we present the results as a structured KG.

5 Conclusion

This paper studies the problem of constructing knowledge graphs using the knowledge embedded in language models. We propose a two-stage approach MAMA to first match the facts with the internal knowledge of LM, then map the matched facts to produce a knowledge graph. The resulting open knowledge graphs are of promising quality and extend the schema of the existing knowledge graphs. In the future, we plan to continue improving both the quality and scale of the constructed knowledge graphs via applying deeper language models (e.g., GPT-3) to larger corpora (e.g., Common Crawl).

REFERENCES

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*, 2016.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *ACL*, pp. 344–354, 2015.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pp. 722–735. 2007.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pp. 1247–1250, 2008.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, number 3, 2010.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W Cohen. Differentiable reasoning over a virtual knowledge base. *arXiv preprint arXiv:2002.10640*, 2020.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pp. 601–610, 2014.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *EMNLP*, pp. 1535–1545, 2011.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, pp. 28–61, 2013.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *TACL*, pp. 521–535, 2016.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *NIPS*, pp. 14014–14024, 2019.

Feng Niu, Ce Zhang, Christopher Ré, and Jude Shavlik. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(3):42–73, 2012.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, (8):9, 2019.

Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. Open language learning for information extraction. In *EMNLP*, pp. 523–534, 2012.

Valentin I Spitkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. 2012.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW*, pp. 697–706, 2007.

Mihai Surdeanu. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. *TAC*, pp. 2, 2013.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017.

Jesse Vig. Visualizing attention in transformerbased language models. *arXiv preprint arXiv:1904.02679*, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pp. 19–27, 2015.