mOKB6: A Multilingual Open Knowledge Base Completion Benchmark

Shubham Mittal[α†] **Keshav Kolluru**[β†] **Soumen Chakrabarti**[γ] **Mausam**[α]
[α] Indian Institute of Technology Delhi
[β] KnowDis AI, New Delhi
[γ] Indian Institute of Technology Bombay
shubhamiitd18@gmail.com, keshav.kolluru@gmail.com
soumen@cse.iitb.ac.in, mausam@cse.iitd.ac.in

Abstract

Automated completion of open knowledge bases (Open KBs), which are constructed from triples of the form (*subject phrase*, *relation phrase*, *object phrase*), obtained via open information extraction (Open IE) system, are useful for discovering novel facts that may not be directly present in the text. However, research in Open KB completion (Open KBC) has so far been limited to resource-rich languages like English. Using the latest advances in multilingual Open IE, we construct the first multilingual Open KBC dataset, called mOKB6, containing facts from Wikipedia in six languages (including English). Improving the previous Open KB construction pipeline by doing multilingual coreference resolution and keeping only entity-linked triples, we create a *dense* Open KB. We experiment with several models for the task and observe a consistent benefit of combining languages with the help of shared embedding space as well as translations of facts. We also observe that current multilingual models struggle to remember facts seen in languages of different scripts.[1]

1 Introduction

Open information extraction (Open IE) systems (Mausam, 2016) such as ReVerb (Etzioni et al., 2011) and OpenIE6 (Kolluru et al., 2020) can extract triples, or *facts*, of the form (*subject phrase*, *relation phrase*, *object phrase*), which can be denoted as (s, r, o), from text (e.g., Wikipedia articles) without using any pre-defined ontology. Open knowledge base (Open KB) is constructed using these Open IE triples where the subject phrases and object phrases are nodes and relation phrases are labels on edges connecting the nodes in the graph. Open knowledge base completion (Open KBC) is the task of discovering new links between nodes using the graph structure of the Open KB. Knowledge graph embedding (KGE) models are typically used for the Open KBC task, where they are asked to answer questions of the form $(s, r, ?)$ and $(?, r, o)$.

Research in Open KBC has been restricted to English (Vashishth et al., 2018) due to lack of Open KBs in other languages. We aim to study multilingual Open KBC, with the motivation that the information available in high resource languages like English may help when inferring links in Open KBs that use low resource languages like Telugu. Moreover, intuitively, if all the information in different languages can be pooled together, then it may help the model learn better, and allow information flow across Open KBs in different languages.

We design the first multilingual Open KB construction pipeline (shown in Figure 1) using a multilingual Open IE system, GEN2OIE (Kolluru et al., 2022). We find that coreference resolution is missing in existing Open KB construction (Gashteovski et al., 2019) but is important for increasing the coverage of facts (as described in Figure 4). We re-train a recent coref model (Dobrovolskii, 2021) using XLM-R (Conneau et al., 2020) as the underlying multilingual encoder and add it to our pipeline. For constructing a high quality test set, we use 988 manually verified facts in English. For extending to other languages, we automatically translate English facts. The dataset thus constructed, called mOKB6, contains 42K facts in six languages: English, Hindi, Telugu, Spanish, Portuguese, and Chinese.

We report the first baselines for multilingual Open KBC task. We find that they are able to benefit from information in multiple languages when compared to using facts from a single language. Translations of Open KB facts also help the models. However, we notice that although the multilingual KGE models learn facts in a particular language, they struggle to remember the same fact, when queried in another language with different script.

† Major part of work done as students at IIT Delhi.
[1] Dataset and code released at github.com:dair-iitd/mokb6

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 201–214

2 Related Work

Multilingual Open KBC datasets are absent in literature to the best of our knowledge, although multiple English Open KBC datasets are available. OLPBench (Broscheit et al., 2020), derived from OPIEC (Gashteovski et al., 2019), is a large-scale Open KBC dataset that contains 30M triples and is constructed from English Wikipedia using MinIE system (Gashteovski et al., 2017). The evaluation data contains 10K triples randomly sampled from 1.25M *linked* triples. ReVerb45K (Vashishth et al., 2018) and ReVerb20K (Galárraga et al., 2014) are smaller Open KBC datasets constructed from Clueweb09 corpus[2] using ReVerb Open IE system (Fader et al., 2011). Both the datasets keep only those tuples in which both the *subject phrase* and *object phrase* link to a finite set of Freebase entities.

Multilingual Open IE (mOpenIE) systems like GEN2OIE (Kolluru et al., 2022) and Multi^{2}OIE (Ro et al., 2020) enable extracting facts from multiple languages. We use the GEN2OIE model for constructing mOKB6 dataset as it is trained with language-specific facts transferred from English, while Multi^{2}OIE relies on zero-shot transfer for languages other than English.

Knowledge Graph Embedding (KGE) Models: Conventional KGE models like TransE (Bordes et al., 2013), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2018), and TuckER (Balazevic et al., 2019) have been used for Open KBC task (Gupta et al., 2019; Broscheit et al., 2020; Chandrahas and Talukdar, 2021; Kocijan and Lukasiewicz, 2021). Given a triple (s, r, o), these models encode the *subject phrase*, *relation phrase*, and *object phrase* from free text, and pass the encodings to a triple-scoring function, which is optimized using binary cross entropy loss. ComplEx has also been used for multilingual closed KBC task (Chakrabarti et al., 2022).

Pretrained language models like BERT (Devlin et al., 2019) have been used in KGE models for the KBC task (Lovelace and Rosé, 2022; Lv et al., 2022; Chandrahas and Talukdar, 2021; Kim et al., 2020). SimKGC (Wang et al., 2022) is the state of the art KGE model on closed KBC task. It computes the score of a triple (s, r, o) as the cosine similarity of the embeddings of $(s; r)$ and (o), computed using two separate pretrained BERT models without any weight sharing.

[2] http://www.lemurproject.org/clueweb09.php/

3 Dataset Curation

We aim to construct a *dense* multilingual Open KB that maximizes the information about a given real-world entity, which may be represented as multiple nodes across languages. Therefore, we consider those Wikipedia articles[3] that are available in six languages: English, Hindi, Telugu, Spanish, Portuguese, and Chinese[4]. This will also help the model learn from facts in high resource language like English and answer queries in low resource language like Telugu. We work with 300 titles randomly sampled from the ones common among all six languages (found using MediaWiki-Langlinks (MediaWiki, 2021)). Thus, we extract facts from 6×300 Wikipedia articles. We discuss the three stages of our pipeline below.

Stage 1 We first process each Wikipedia article through a coreference resolution system. Although language-specific end-to-end neural coref models have been developed (Žabokrtský et al., 2022; Xia and Van Durme, 2021), multilingual models that work on all our languages of interest are absent in the literature. Therefore, we retrain wl-coref (Dobrovolskii, 2021) with XLM-R (Conneau et al., 2020) on the English training data (available in OntoNotes (Weischedel et al., 2013)) that can work zero-shot for other languages.

Coref models detect and cluster mentions, but do not identify a canonical cluster name, which is needed for standardizing all the mentions in the cluster. To find cluster names, entity linking systems such as mGENRE (De Cao et al., 2022) or Wikipedia hyperlinks can be used. However, we found that they result in low recall, particularly for low resource languages. Thus, we employ a heuristic to find the cluster name and replace each of the coreferent mentions with it. The score for each mention is represented by a tuple, computed as: Score(mention phrase) = (#proper nouns, #nouns, #numerals, # adjectives, #pronouns, #verbs). The tuple is ordered according to the importance of each field (POS tags) for the cluster name, which is determined empirically. Two tuples are compared index-wise with higher priority given to lower indices to determine the best scoring mention that is chosen as the canonical name (Table 1).

Stage 2 We use GEN2OIE to extract Open IE triples from the coreference resolved sentences.

[3] Wikidump of April 02, 2022

[4] languages are chosen to match availability of Gen2OIE

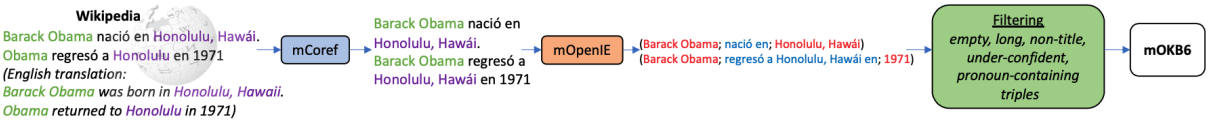

Figure 1: Our three-staged multilingual Open KB construction pipeline for mOKB6. mCoref is multilingual coreference resolution system, having XLM-R (Conneau et al., 2020) encoder based wl-coref (Dobrovolskii, 2021), and mOpenIE is multilingual open information extraction system, consisting of GEN2OIE (Kolluru et al., 2022).

Mentions	Scores	Cluster Name
Barack Obama	**(2,0,0,0,0,0)**	
Obama	(1,0,0,0,0,0)	Barack Obama
He	(0,0,0,0,1,0)	

Table 1: Parts of speech tags are used to find the canonical name of the coreferent cluster of entity mentions.

Stage 3 Similar to Gashteovski et al. (2019), we apply various filters to remove *noisy* triples that have empty or very long arguments, or have less confidence than 0.3 (as assigned by GEN2OIE). We further only keep triples that have the article's title as either the *subject phrase* or *object phrase*, to avoid generic or specific triples, valid only in the particular context. Examples of *contextual* triples (Choi et al., 2021) are discussed in Appendix E. See Appendix A for further data curation details.

These automatically extracted triples form the train set of mOKB6. To form a high quality test set in six languages with limited access to experts in all languages, the test set is created in a semi-automatic way. We sample 1600 English triples from the train set (which are subsequently filtered) and manually remove noisy triples. We use inter-annotation agreement between two annotators to check if they both agree that the given triple is noisy or clean. With an agreement of 91%, we retain 988 English triples, which we automatically translate to the other five languages. As illustrated in Figure 2, to translate a triple, we convert it to a sentence after removing tags and use Google translate[5] for translating the triple-converted sentence to the remaining five languages. We observed high quality of translated triples, with 88% satisfactory translations as determined by native-speakers of three languages on a set of 75 translated triples. To get the Open IE *subject phrase*, *relation phrase* and *object phrase* tags, we project the labels from the original English triple to the translated sentence using word alignments (Kolluru et al., 2022). Finally, we are left with 550 triples in each language after removing examples where some labels could not be aligned. We use these 6×550 triples as the test sets. The train and dev sets are created from the remaining triples in each language such that the dev set has 500 randomly sampled triples (Table 2).

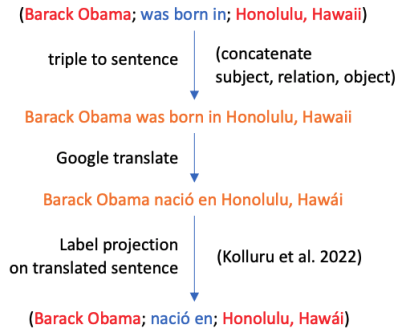

Figure 2: Method to translate Open IE triple using Google translate, and followed by label projection using word alignments (Kolluru et al., 2022).

We analyse the entity overlap across languages and find that on an average, a test entity (which is present in either the *subject phrase* or *object phrase* of a test tuple) is present 17.73 times in English, 0.94 times in Hindi, 0.47 times in Telugu, 2.33 times in Spanish, 1.69 times in Portuguese, and 1.45 times in Chinese train set.

Our construction pipeline improves over OPIEC in three ways: (1) we use a multilingual Open IE system, instead of an English-specific Open IE system like in OPIEC, enabling us to curate Open KBs in many languages, (2) we add a multilingual coreference resolution system in our pipeline, and (3) the English test triples are manually verified. Further, we manually evaluate and review the noise at each step of data curation in Section 4.

	En	Hi	Te	Es	Pt	Zh
#entity	20637	4625	3972	5651	5304	5037
#relation	7870	2177	1907	2823	2644	2325
#train	20195	2786	1992	3966	3528	3420

Table 2: Statistics of individual Open KBs in mOKB6 in English (En), Hindi (Hi), Telugu (Te), Spanish (Es), Portuguese (Pt), and Chinese (Zh). The dev and test set for each Open KB contain 500 and 550 triples each.

[5] https://translate.google.co.in/

4 Noise Evaluation

Curating an Open KB involves various stages and each stage induces its noise in the construction pipeline (Gashteovski et al., 2019). We manually evaluate the noise induced at each stage of our pipeline (Figure 1) and discuss the same in this section. We ask native speakers of four (out of six) languages - English, Hindi, Telugu, and Chinese to assess the output quality, or precision, of each stage as discussed below.

In the first stage, we assess the performance of the coreference resolution system over Wikipedia articles. We find a high precision of 95.5% in coref's mention clustering and 89.82% accuracy in finding canonical cluster name (using the heuristic illustrated in Table 1), computed over 40 randomly sampled coref clusters (10 in each language).

For evaluating the Open IE system, GEN2OIE, in the second stage, we mark an extraction of a sentence as correct if it has syntactically correct arguments and it is coherent with the sentence. We get an average precision of 63.4% on 80 extractions (20 in each language).

We evaluate the triples, or Open KB facts, at the last stage after passing through various noise-removing filters. Note that these triples also form the train set (and dev set) in mOKB6 dataset. We mark triples as correct when they contain real-world entities, and also, factual information about them. If the triple is very generic or contextual (see Appendix E), it is marked as incorrect. We find the train (and dev) set quality to be 69.3%, averaged over 80 triples in four languages.

5 Experiments

Our experimental study on multilingual open KBC task investigates the following research questions:

1. Does the KGE model benefit from facts in different languages? (Section 5.1)
2. Can translation help transfer among languages? (Section 5.2)
3. Does the KGE model remember facts seen across different languages? (Section 5.3)

We use SimKGC model (Wang et al., 2022) with pretrained mBERT initialization to run our experiments, after comparing with recent KGE models (Appendix C). For evaluation, we use three metrics — hits at rank 1 (H@1), hits at rank 10 (H@10), and mean reciprocal rank (MRR). The formal definitions of them are provided in Appendix B. We discuss further model training details in Appendix D.

5.1 Training on Multilingual Facts

We train and compare monolingual model, called MONO, with multilingual models, UNION and UNION w/o En. In MONO, we train one model for each language using its respective Open KB, whereas in UNION, a single model is trained on six languages' Open KBs together. UNION outperforms MONO in all languages by an average of 4.6% H@10 and 2.8% MRR (see Table 3), which provides evidence of information flow across languages and the model benefits from it.

To check the extent of flow from (high-resource) English to the other languages, we also train on the five languages except English, which we call UNION w/o En. We find UNION w/o En also outperforms MONO by 2.7% H@10 and 1.2% MRR over the five languages, hinting that interlingual transfer is more general and pervasive.

5.2 Open KB Facts Translation

Apart from relying only on multilingual transfer in the embedding space, we analyse the effect of using translated triples in the training of the KGE model. We translate the English training triples[6] to the other five languages (Section 3) and train monolingual models using only the translated triples (TRANS). To leverage facts present in each language's Open KB, we make MONO+TRANS, where we add language-specific MONO data to the translated triples. Table 3 shows that MONO+TRANS is better than MONO by a large margin of 15.5% H@1, 29.2% H@10, and 20.0% MRR, averaged over five languages. Also, MONO+TRANS improves over TRANS by 2.1% H@10 and 2.0% MRR, showcasing the importance of facts in each language's Open KBs.

To effectively gain from transfer in both the embedding space as well as translation, we introduce UNION+TRANS. We train one model for each language, on the combination of UNION triples and the translated train triples from English Open KB to that language. UNION+TRANS is better than UNION by 25.9% H@10 and 18.4% MRR. This suggests that the model is able to benefit from English facts when they are translated to the query language, unlike in UNION where the English facts are present only in English.

[6] English source achieved the best translation quality.

	English (En)			Hindi (Hi)			Telugu (Te)			Spanish (Es)			Portuguese (Pt)			Chinese (Zh)		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MONO	14.8	38.7	22.8	3.0	14.8	7.2	1.5	8.1	3.9	6.4	23.7	12.3	6.3	21.7	11.4	2.4	13.1	6.2
UNION w/o En	5.7	21.5	10.9	2.9	15.4	7.4	1.8	10.2	4.9	8.1	27.8	14.5	6.7	26.1	12.9	3.2	15.5	7.5
UNION	**16.7**	**40.8**	**24.8**	3.6	16.6	8.1	1.5	9.3	4.5	10.6	32.2	17.6	9.7	29.3	16.6	4.0	18.8	8.9
TRANS	-	-	-	20.5	47.6	29.7	8.7	28.7	15.5	23.2	50.6	32.4	20.5	50.7	30.5	14.0	39.4	22.5
MONO+TRANS	-	-	-	20.2	45.4	28.4	14.3	**38.5**	22.2	23.5	51.5	32.9	21.4	48.9	30.7	**17.9**	43.2	**26.6**
UNION+TRANS	-	-	-	**23.3**	**49.7**	**32.3**	**15.1**	**38.5**	**23.1**	**23.9**	**52.4**	**33.4**	**23.5**	**52.1**	**33.1**	16.9	**43.6**	26.0

Table 3: Performance (%) of SimKGC model on mOKB6 dataset, comprising of Open KBs in six languages. MONO, TRANS, and MONO+TRANS are monolingual models trained only on facts of one language whereas UNION, UNION w/o En, and UNION+TRANS are multilingual models trained with facts from multiple languages. All reported numbers are an average of three runs using different seeds. Best scores are highlighted in bold.

5.3 Cross-lingual Memorization

Pretrained multilingual language models such as mBERT have demonstrated strong cross-lingual transfer capabilities (Wu and Dredze, 2019). We investigate cross-lingual memorization of the KGE model by showing facts in one language and querying the same facts in other five languages. For each language, L, we take the UNION model and train it further on the test set of that language's Open KB, which we call MEMORIZE$_L$ model. Then, we test each MEMORIZE$_L$ model on the six test sets. Since the test sets (in mOKB6 dataset) of the different languages contain the same facts, this experiment allows us to investigate cross-lingual memorization. We provide the H@10 scores of MEMORIZE models in Figure 3 and the performance on other metrics (H@1 and MRR) is reported in Table 7.

The model achieves at least 97% H@10 when tested on the language used for training (diagonal). We observe that there is relatively good cross-lingual memorization among languages that share the same script (Latin in English, Spanish, and Portuguese), but the model struggles to remember facts when seen in languages of different scripts. Many entities look similar in shared scripts, possibly leading to better information transfer. For example, the MEMORIZE$_{En}$ achieves H@10 of 50.7% in Spanish (Es) compared to 22.3% in Chinese (Zh) and 11% in Telugu (Te).

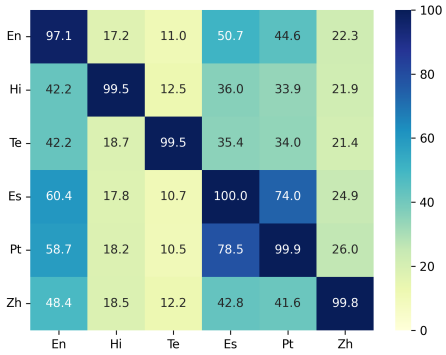

Figure 3: Performance (H@10) of MEMORIZE models. Row L shows the performance of MEMORIZE$_L$ model across the test sets of all languages (columns). For example, the performance of MEMORIZE$_{En}$ when tested on English (En) is 97.1% H@10, and MEMORIZE$_{En}$ when tested on Spanish (Es) gives 50.7% H@10. We find relatively good cross-lingual transfer among languages that use same script (Latin in English, Spanish and Portuguese) compared to those using different scripts (English, Hindi, Telugu and Chinese).

6 Conclusion and Future Work

We create and release the mOKB6 dataset, the first multilingual Open Knowledge Base Completion dataset with 42K facts in six languages: English, Hindi, Telugu, Spanish, Portuguese, and Chinese. Its construction uses multilingual coreference resolution, entity-mention cluster naming, multilingual open information extraction and various filtering steps to improve the quality of the extracted facts. We also report the first baselines on the task using the existing state of the art KGE models trained with facts from different languages using various augmentation strategies.

Our work opens many important research questions: (1) Can we develop better strategies to combine facts in different languages? (2) Can we build models that achieve strong information transfer across unrelated languages with same or different scripts? (3) Can we train the neural model to ignore contextual triples (Appendix E), thus improving overall performance? and (4) Can tying the same entities across various languages help the model generalize better? We leave these questions to be addressed in future work.

7 Acknowledgements

Keshav was supported by TCS Research Fellowship during his PhD. Mausam is supported by grants from Huawei, Google, Verisk and IBM, and a Jai Gupta Chair Fellowship. He also acknowledges Google and Yardi School of AI travel grants. Soumen is partly supported by a Jagadish Bose Fellowship and a grant from Cisco. We thank IIT Delhi HPC facility for compute resources.

8 Limitations

Although multilingual, the constructed open KB is limited to the sampling of the chosen six languages. We do not know how well the system will generalize to various language families that have not been considered here. Further, even among the languages considered, the performance of even the best-performing systems, as measured through H@1 is still in the low 20's. Therefore the models are not yet ready to be deployed for real-world applications.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2308, Online. Association for Computational Linguistics.

Soumen Chakrabarti, Harkanwar Singh, Shubham Lohiya, Prachi Jain, and Mausam . 2022. Joint completion and alignment of multilingual knowledge graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11922–11938, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

. Chandrahas and Partha Talukdar. 2021. OKGIT: Open knowledge graph link prediction with implicit types. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2546–2559, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL Conference*, pages 8440–8451.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *IJCAI*

2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011, pages 3–10. IJCAI/AAAI.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. New York, NY, USA. Association for Computing Machinery.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2019. Opiec: An open information extraction corpus. In *Proceedings of the Conference on Automatic Knowledge Base Construction (AKBC)*.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. CaRe: Open knowledge graph embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 378–388, Hong Kong, China. Association for Computational Linguistics.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Vid Kocijan and Thomas Lukasiewicz. 2021. Knowledge base completion meets transfer learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761, Online. Association for Computational Linguistics.

Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, and Mausam . 2022. Alignment-augmented consistent translation for multilingual open information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, Dublin, Ireland. Association for Computational Linguistics.

Justin Lovelace and Carolyn Rosé. 2022. A framework for adapting pre-trained language models to knowledge graph completion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5937–5955, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3570–3581, Dublin, Ireland. Association for Computational Linguistics.

Mausam. 2016. Open information extraction systems and downstream applications. In *International Joint Conference on Artificial Intelligence*.

MediaWiki. 2021. Api:langlinks — mediawiki,. [Online; accessed 02-April-2022].

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multi^2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1107–1117, Online. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA. PMLR.

Shikhar Vashishth, Prince Jain, and Partha Talukdar. 2018. CESI: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1317–1327, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, and Michelle Franchini. 2013. Ontonotes release 5.0. In *Linguistic Data Consortium, Philadelphia, PA*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Patrick Xia and Benjamin Van Durme. 2021. Moving on from OntoNotes: Coreference resolution model transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5241–5256, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, Daniel Zeman, and Yilun Zhu. 2022. Findings of the shared task on multilingual coreference resolution. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 1–17, Gyeongju, Republic of Korea. Association for Computational Linguistics.

mOKB6: A Multilingual Open Knowledge Base Completion Benchmark
(Appendix)

A Dataset Curation

As discussed in Section 3, we construct mOKB6 dataset in three stages after extracting the Wikipedia articles (using WikiExtractor[7]) from the Wikidump of April 02, 2022. We run our construction pipeline (as shown in Figure 1) for all six languages on a single V100 (32 GB) GPU, which required 14 hours of computation to create mOKB6 dataset.

In the first stage, we keep the sentences containing at least 6 and at most 50 tokens since we find that most of the short sentences are headings or sub-headings present in Wikipedia articles, and very long sentences can't be input to GEN2OIE (in second stage) due to maximum sequence length constraint of 1024 in mT5 (Xue et al., 2021) based GEN2OIE. This filtering step discards 18.9% of sentences on an average in all six languages. We use Stanza (Qi et al., 2020) to perform sentence- and word-segmentation on Wikipedia articles in all six languages. After filtering the sentences, the articles are processed for coreference resolution using XLM-R (Conneau et al., 2020) encoder based wl-coref (Dobrovolskii, 2021), followed by replacing the coreferent cluster mentions with their canonical cluster name using the heuristic discussed in Section 3.

In the second stage, the coreference resolved articles are passed through GEN2OIE to get the Open IE triples. The confidence scores for these triples are computed using label rescoring, for which we refer the readers to Kolluru et al. (2022) for more details.

Finally, in the last stage, we apply various filters, adapted from Gashteovski et al. (2019), to remove triples that are of no interest to Open KBC task, like the triples: (1) having any of its argument or relation empty, (2) containing more than 10 tokens in any of its arguments or relation, (3) having confidence score less than 0.3, (4) containing pronouns (found using Stanza) in its arguments, (5) having same subject and object (i.e. self loops), and (6) that are duplicates. These filters keep 91.6% of the triples obtained from stage 2 in all six languages.

[7] https://github.com/samuelbroscheit/wikiextractor-wikimentions

Further in the last stage, in order to create a *dense* Open KB containing minimum noise and maximum facts about the entities, we keep the triples having the Wikipedia article's title as either the *subject phrase* or *object phrase* and discard the rest. We do this by finding all the coreference clusters (of entity mentions) that contain the titles, then get the entities, or cluster names, of those clusters using the heuristic discussed in section 3, and keep those triples that contain these cluster names. This filtering step retains 23.6% of the triples.

B Metrics

We follow the previous works (Wang et al., 2022) on the evaluation methodology of Open KBC task and apply it to the multilingual Open KBC task, containing facts in multiple languages. Given an Open KB, containing a finite set of entities and open relations, the KGE model answers forward and backward queries of the form $(s, r, ?)$ and $(?, r, o)$ respectively. The model ranks all the entities based on their correctness with, say, s and r in the forward query. Further, the evaluation is in *filtered* setting, where the other known correct answers, apart from o, are removed from rank list.

The commonly used evaluation metrics are hits at rank N (H@N), where N is a natural number, and mean reciprocal rank (MRR). Suppose, the model ranks o at R among all entities. Then, H@N measures how many times R is less than or equal to N. MRR is the average of reciprocal ranks ($\frac{1}{R}$). Both, H@N and MRR, are computed as average over both forms of queries over the full test set.

C Knowledge Graph Embedding Models

SimKGC (Wang et al., 2022) is a text-based KGE model that uses two unshared pretrained BERT models (Devlin et al., 2019) for encoding *(subject phrase; relation phrase)* and *object phrase* separately. GRU-ConvE (Kocijan and Lukasiewicz, 2021) encodes both the *relation phrase* and *argument phrase* from their surface forms using two unshared GRU (Cho et al., 2014). CaRe (Gupta et al., 2019) learns separate embeddings for each *argument phrase* and uses a bi-directional GRU to encode the *relation phrase* from its surface form. Both, GRU-ConvE and CaRe, are initialised with Glove embeddings (Pennington et al., 2014).

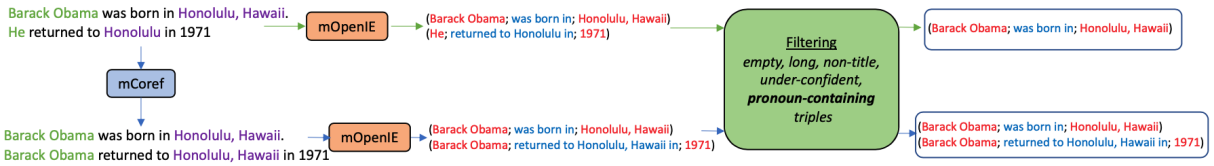

Figure 4: Previous Open KB construction pipelines like Gashteovski et al. (2019) (shown by green arrows) lack coreference resolution system, which result in filtering *important* facts like (*Barack Obama*; *returned to Honolulu, Hawaii in*; *1971*). Our pipeline (shown by blue arrows) increases the *coverage* of facts due to mCoref system.

To choose the best model for our experiments (Table 3, Figure 3), we train the recent knowledge graph embedding (KGE) models — CaRe,, GRU-ConvE and SimKGC on the English Open KB in mOKB6. We report performance in Table 4 using the three metrics: hits at rank 1 (H@1), hits at 10 (H@10), and mean reciprocal rank (MRR). We find that SimKGC with BERT encoder outperforms the other two models.

	H@1	H@10	MRR
CaRe	6.6	11.3	8.3
GRU-ConvE	12.4	27.8	17.8
SimKGC (BERT)	**16.1**	**40.0**	**24.3**
SimKGC (mBERT)	**14.8**	**38.7**	**22.8**
SimKGC (XLM-R)	13.8	35.8	21.3

Table 4: Performance (%) of the KGE models on the English test set in mOKB6 dataset. The reported numbers are an average of three runs using different seeds.

Since BERT supports only English language, we replace BERT in SimKGC with multilingual pre-trained language models like mBERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020), to extend SimKGC model to other languages. We find in Table 4 that SimKGC with mBERT is better than with XLM-R by 2.9% H@10 and 1.5% MRR, possibly because mBERT (and mOKB6) uses Wikipedia while XLM-R uses CommonCrawl (Wenzek et al., 2020) during pre-training. Thus, we use SimKGC with mBERT as the underlying encoder to run our experiments for all the languages.

D KGE Model Training Details

We use the code from official repositories of the KGE models — SimKGC (Wang et al., 2022), GRU-ConvE (Kocijan and Lukasiewicz, 2021), and CaRe (Gupta et al., 2019) for our experiments. The models are trained using Adam optimizer (Kingma and Ba, 2015) on a single A100 (40 GB) GPU with three different random seeds and we report the average of three evaluation runs.

We do not perform hyperparameter search trials, except for batch size, and use the default hyperparameters from the respective codes of KGE models (see Table 5). We use early stopping to find the best model checkpoints based on HITS@1. The dev set is different for each baseline: MONO, TRANS, MONO+TRANS, and UNION+TRANS use individual language's dev set, whereas UNION w/o En and UNION use the English dev set. We report the performance of baseline models on the dev sets in Table 9 and Table 10.

Hyperparameter	SimKGC	GRU-ConvE	CaRe
#epochs	100	500	500
#patience epochs	10	10	10
learning rate	3e-5	3e-4	1e-3
dropout	0.1	0.3	0.5
batch size	256	1024	128
additive margin	0.02	N/A	N/A

Table 5: Hyperparameters of the KGE models.

We provide the number of trainable parameters of each KGE model in Table 6. Based on the batch size and model size, different experiments consume different GPU hours. To train on English Open KB (in mOKB6 dataset), CaRe and GRU-ConvE models took 2.5 hours and 0.5 hours, respectively, whereas SimKGC takes nearly 1 hour of GPU time.

KGE model	#trainable parameters
CaRe	12,971,423
GRU-ConvE	12,085,523
SimKGC (BERT)	216,620,545
SimKGC (mBERT)	355,706,881
SimKGC (XLM-R)	1,119,780,865

Table 6: Number of trainable parameters in the KGE models.

	English			Hindi			Telugu			Spanish			Portuguese			Chinese		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
English	68.4	97.1	78.8	3.4	17.2	8.3	1.6	11	5	17.8	50.7	28.6	17	44.6	26	5.4	22.3	11.1
Hindi	19	42.2	26.7	80.6	99.5	88.3	2.4	12.5	5.9	12.3	36	19.9	12.3	33.9	19.7	5.3	21.9	10.8
Telugu	19.5	42.2	27.2	4.3	18.7	9.4	74.4	99.5	84.2	10.9	35.4	18.9	10.7	34	18.5	4.7	21.4	10.1
Spanish	27.9	60.4	38.8	4.1	17.8	8.9	1.8	10.7	5.1	84	100	90.3	37.6	74	50.1	6.5	24.9	12.8
Portuguese	27.8	58.7	38.2	4.4	18.2	9.3	1.7	10.5	5.1	41.5	78.5	53.6	84.2	99.9	90.8	6.6	26	13.2
Chinese	22.1	48.4	30.6	3.5	18.5	8.8	1.8	12.2	5.4	14.8	42.8	24.2	15.7	41.6	24.1	81.6	99.8	89.2

Table 7: Performance (%) of the six MEMORIZE models, which have been trained on each language's test set and tested on all the test sets in mOKB6 dataset.

E Contextual Triples

Open IE triples are of various kinds and not all of them can be used for Open KBC task. Various filtering steps are used to remove some of these in data curation (Section 3). We define *contextual* triples as another kind of noisy triples, which are specific to, and are not interpretable out of, the context of text from which they are extracted.

(*Max Born*; *continued*; *scientific work*)
(*Robb Gravett*; *won*; *the championship*)
(*George Herbert Walker Bush*; *was*; *out of touch*)
(*Christianity*; *is*; *dominant*)

Table 8: Examples of contextual triples.

From the first two triples in Table 8, it is unclear which scientific work *Max Born* continued, or which championship *Robb Gravett* has won. The last two triples are too specific to the context and contain no factual information.

	English (En)			Hindi (Hi)			Telugu (Te)			Spanish (Es)			Portuguese (Pt)			Chinese (Zh)		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MONO	16.2	38.7	23.9	18.2	39.4	25.9	8.5	20	12.5	17.3	36.6	23.7	17.6	39.6	25.3	10.8	31.9	17.8
TRANS	-	-	-	8.1	23.7	13.5	3.3	15.4	7.5	12.9	33.6	20.3	12.6	37.2	20.6	5	20.8	10.3
MONO+TRANS	-	-	-	20.8	43.2	28.6	7.8	24.8	13.4	20.2	46	28.8	21	45.9	29.2	10.6	30.1	16.7
UNION	19.9	39.6	26.4	14.5	38.2	22.4	5.9	20	10.6	19.8	43.2	27.9	19.7	43.8	28	11.2	33	18.8
UNION w/o En	5.8	19.5	10.6	15.4	39.3	23.3	6.3	20.5	11.1	19.4	41.6	26.4	16.9	42.9	25.9	11.3	33	18.4
UNION+TRANS	-	-	-	20.8	44.9	28.8	7.3	27.1	14	21.4	45.3	29.6	19.4	49.1	29.1	6.9	31	15.1

Table 9: Performance (%) of SimKGC on the dev sets (of mOKB6 dataset) in six languages.

	H@1	H@10	MRR
CaRe	7.1	11.1	8.5
GRU-ConvE	16.8	31.5	22.1
SimKGC (BERT)	20.3	40.1	27.1
SimKGC (mBERT)	16.2	38.7	23.9
SimKGC (XLM-R)	17	36.6	23.2

Table 10: Performance (%) of the KGE models on dev set of English Open KB in mOKB6 dataset.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☒ A2. Did you discuss any potential risks of your work?
There are no potential risks of our work to our knowledge.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3,4

☑ B1. Did you cite the creators of artifacts you used?
3,4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Abstract

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix D

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix A, D

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

3

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
4

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.