Unsupervised Knowledge Graph Generation Using Semantic Similarity Matching

Lixian Liu[a], **Amin Omidvar**[a], **Zongyang Ma**[a], **Ameeta Agrawal**[b], **Aijun An**[a]
[a]Department of Electrical Engineering and Computer Science, York University, Canada
[b]Department of Computer Science, Portland State University, USA
lixian@my.yorku.ca, omidvar@yorku.ca, mzyone@gmail.com
ameeta@cs.pdx.edu, aan@eecs.yorku.ca

Abstract

Knowledge Graphs (KGs) are directed labeled graphs representing entities and the relationships between them. Most prior work focuses on supervised or semi-supervised approaches which require large amounts of annotated data. While unsupervised approaches do not need labeled training data, most existing methods either generate too many redundant relations or require manual mapping of the extracted relations to a known schema. To address these limitations, we propose an unsupervised method for KG generation that requires neither labeled data nor manual mapping to the predefined relation schema. Instead, our method leverages sentence-level semantic similarity for automatically generating relations between pairs of entities. Our proposed method outperforms two baseline systems when evaluated over four datasets.

1 Introduction

A knowledge graph (KG) is a directed labeled graph in which nodes represent entities and edges are labeled by well-defined relationships between entities. Formally, given a set E of entities and a set R of relations, a knowledge graph is a set T of triples, where $T \subseteq E \times R \times E$. A triple $t \in T$ can be expressed as (e_h, r, e_t), where $e_h \in E$, $r \in R$, $e_t \in E$, and e_h and e_t are referred to as the head entity and the tail entity, respectively. As a structured representation of world knowledge, knowledge graphs have been used in a number of applications such as Web search (Singhal, 2012; Wang et al., 2019a), question answering (Huang et al., 2019) and recommender systems (Wang et al., 2019b).

Knowledge graphs can be constructed automatically from text. Most of the automatic KG generation methods are supervised or semi-supervised, where a large set of labeled data is required to train a KG generation model (e.g., PCNN (Zeng et al., 2015), OLLIE (Schmitz et al., 2012), ReVerb (Fader et al., 2011)). However, creating labeled data is labor-intensive and the generated graph is limited to the specific domain of the training corpus. In addition, supervised methods can only extract a predefined set of relations occurring in the training data and the model needs to be re-trained to work with other new relation schemas.

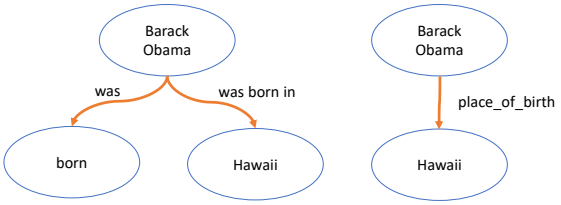

Figure 1: A KG generated using Stanford OpenIE (left) and our method (right) for the input sentence "*Barack Obama was born in Hawaii*".

Unsupervised KG models (e.g., Stanford OpenIE (Angeli et al., 2015)), on the other hand, do not need labeled training corpus. They often use syntactic parsing and a set of rules to extract relationships between two entities in a sentence. Although not normally confined to a predefined set of relations, too many unuseful or inaccurate relations can be generated. In Figure 1, the left graph presents an example KG using triples generated with Stanford OpenIE (Angeli et al., 2015), while the right graph presents the KG generated using our proposed method, both using the same single input sentence. In addition, in case only relations in a predefined set need to be generated, the unsupervised methods do not normally provide a mechanism to map the extracted relation to a known one in the set of relations

In a project to build knowledge graphs from news articles where no labeled data are given, we propose an unsupervised knowledge graph generation method using semantic similarity (KGSS) that does not need a labeled set of training data nor a complicated set of syntactic rules for KG

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 169 - 179

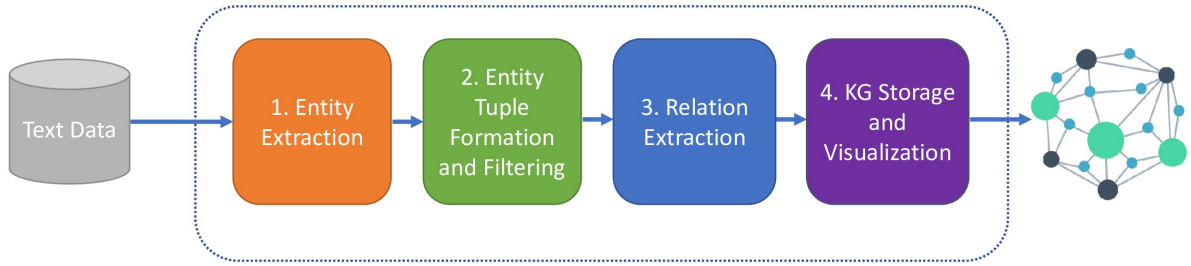

Figure 2: An overview of KGSS, our proposed unsupervised KG generation system.

generation. The method can work with any set of relations that a user prefers, and uses semantic similarity matching to automatically identify the relation between two entities. A salient feature of our method is the use of a pretrained language model (Reimers and Gurevych, 2019) to compute and measure the similarity between the sentence embedding and the embedding of candidate triples formed by the two entities and a candidate relation. The best matching candidate relation is identified as the relation between the two entities.

Since most supervised models underperform in low-resource settings where no or very limited labeled data are provided, our proposed unsupervised approach can extract useful relations from unlabeled data and can also be used to create a labeled data set for distant supervised learning, which can potentially lead to better results. In this paper, we focus on describing and evaluating the unsupervised method.

The contributions of this paper are as follows:

- We propose a novel unsupervised KG generation system that requires no labeled data.
- Our method is flexible and can work with any set of relations. The results of the empirical evaluation (automatic as well as human) demonstrate that our system significantly outperforms two state-of-the-art unsupervised methods for KG generation.
- To facilitate research in KG construction or information extraction from news articles, we develop a new dataset called NewsKG21[1] that was created from recent news articles.

[1]The NewsKG21 dataset and the code for our KG generation and visualization are available under the open source license at https://github.com/lixianliu12/KGSS

2 Related Work

Research on KG construction falls under supervised, semi-supervised, or unsupervised categories. For the supervised methods, we name two of them. Bastos et al. (2021) propose the RECON model to extract relations from a sentence and align them to the KG, using a graph neural network for obtaining the sentence representations. Then a neural classifier is adopted to predict the relation of each entity pair in the sentence. Another supervised learning method for KG construction is SpERT (Eberts and Ulges, 2020), which is a span-based deep learning model with the attention mechanism, targeting to extract entities and relations jointly. Semi-supervised approaches such as ReVerb (Fader et al., 2011), OLLIE (Schmitz et al., 2012), and Stanford OpenIE (Angeli et al., 2015), to name a few, leverage linguistic features (e.g., dependency trees and POS tags) with many human-defined patterns and existing knowledge bases (e.g., Wikidata (Vrandečić and Krötzsch, 2014), DBpedia (Auer et al., 2007)) to extract triples. These systems have a supervision component. For example, Stanford OpenIE uses distant supervision to create a noisy corpus of sentences annotated with relation mentions and train a logistic regression classifier to decide which action to perform on an edge on the parse tree when extracting relations. However, these systems miss many potential triples in a sentence since they use verbs as a signal to identify triples, whereas many relational triples may not be connected with a verb. They also tend to generate redundant triples and require manual mapping of the extracted relations to a fixed relation schema.

The earliest unsupervised approaches (i.e., heuristics approaches) (Suchanek et al., 2007; Auer et al., 2007; Bollacker et al., 2008) were applied to Wikipedia data, building the pioneering Knowledge Graphs (e.g., YAGO, DBpedia, Freebase). However, these approaches leverage additional

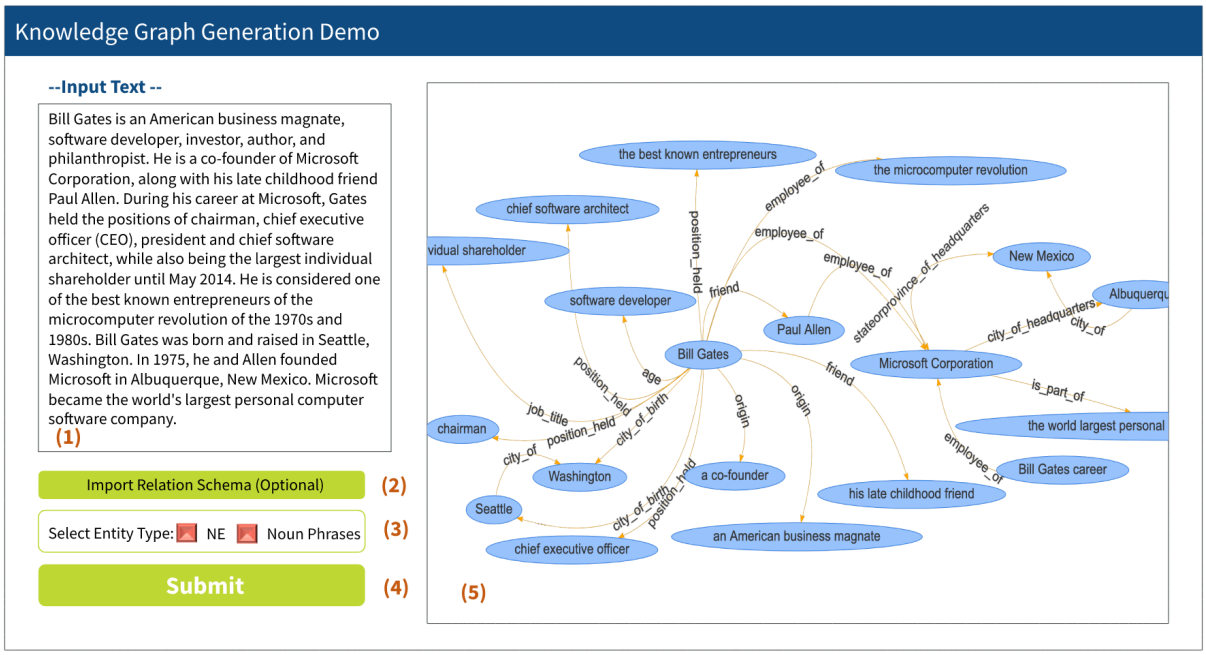

Figure 3: A demo of our system. (1) An input box for users to enter text. (2) A button for users to select their preferred relation schema; if nothing is imported, a default relation schema is used. (3) Users can select the type of entities to be extracted; if nothing is selected, both Named Entity and Noun will be extracted. (4) A submit button. (5) An interactive KG will be generated and visualized where the users can drag the nodes around to modify the presentation of the graph as desired.

knowledge to construct the graph, for example, the Wikipedia hierarchical categories in (Suchanek et al., 2007). Another drawback of these approaches is that they are slow and costly to build the KG. The resultant KGs are also restricted to a specific domain of corpus. MAMA (Wang et al., 2020), an unsupervised KG construction model, uses the attention weight matrices of a pre-trained language model (e.g., BERT (Devlin et al., 2018)) to extract the candidate triples. For mapping the extracted relations to a fixed schema, they follow the method of Stanford OpenIE (Angeli et al., 2015) requiring some manual annotations. Goswami et al. (2020) propose the RE-Flex framework for unsupervised relation extraction, where given a set of relations, each of them is rewritten as a cloze template (e.g., the cloze template of DraftBy is *X was created by Y*, where X and Y denote subject and object respectively.). Then the cloze template is semantically matched with the context (e.g., "Bill Gates founded Microsoft") to determine if the context has the relation or not. Another simliar work is proposed in (Tran et al., 2020) where the importance of the feature ENTITY TYPE for relation extraction is emphasized in their model called EType+. However, the feed-forward neural network classifier which is incorporated in their EType+ model makes their method not entirely unsupervised.

3 Proposed Model: KGSS

Given a document, our system generates a knowledge graph from the document. Figure 2 illustrates an overview of our system, KGSS, which consists of four modules: *entity extraction, entity tuple formation and filtering, relation extraction, and KG storage and visualization*, and Figure 3 illustrates the user interface of our system and visualizes a KG generated given an input paragraph based on a relation schema in TACRED* with 6 additional relations: *loc:province_of, loc:country_of, loc:city_of, org:is_part_of, per:position_held* and *per:friend*. Since our proposed system is unsupervised, it can flexibly work with any user-specified relation schema.

3.1 Entity Extraction

The first step in our system is co-reference resolution, which identifies and replaces different expressions of the same real-world entity with the same expression. We use an end-to-end neural coreference resolution model (Lee et al., 2017) from AllenNLP (Gardner et al., 2018) for this task.

In the second step, our system extracts all entities. We allow the user to specify in the user

interface whether they would like to extract only named entities or also include other noun phrases. A named entity (NE) refers to a real-world object associated with a name, for example - a person, an organization, or a location (e.g., Barack Obama, Apple Inc., New York City). We use a transition-based algorithm (Lample et al., 2016) from the spaCy[2] library to detect all the NEs in a given sentence. There are 18 categories of NEs, such as PER (for person), ORG (for organization), and LOC (for location) in the spaCy *en_core_web_lg* pipeline for the NER task. We keep the NEs in all categories. In addition, if noun phrases are to be included, we extract all noun phrases (also called noun chunks) as candidate entities.

3.2 Entity Tuple Formation and Filtering

After extracting entities, we form a set of entity tuples for each sentence as follows. For each sentence s in the input document, let $E = (e_1, e_2, ..., e_k)$ be the list of identified entities in s, where e_i occurs before e_j in s when $i < j$. The set T of entity tuples for s contains all pairs $\langle e_i, e_j \rangle$ such that e_i occurs before e_j in s, that is, $T = \{\langle e_i, e_j \rangle | i < j\}$. We refer to this tuple formation rule as TF1. Thus, for a sentence containing k extracted entities, there are $\frac{k(k-1)}{2}$ entity tuples in its T. As an example, consider the sentence "*Barack Obama was born in Honolulu and graduated from Columbia University.*". The list of extracted entities is *Barack Obama, Honolulu, Columbia University*, and the set of entity tuples is ⟨Barack Obama, Honolulu⟩, ⟨Barack Obama, Columbia University⟩, and ⟨Honolulu, Columbia University⟩.

However, not all entity tuples lead to generation of good relations between the two entities. Thus, we use some heuristic rules to filter out unpromising tuples. Recall that NEs have categories. We use NE_{PER} to denote an NE in the person category, NE_{ORG} an organization NE, and NE_{LOC} a location NE. In addition, we denote all noun phrases as NE_{NOUN}. Not all the combinations of entities will yield meaningful relations between them. For instance, a location subject is most likely to not have a relation with its non-location object (Wang, 2020). Thus, we leverage the NE types and apply the following rules to keep quality candidate tuples and filter out some invalid ones: Rule TF2: keep all the tuples whose head entity is a NE_{PER}, a NE_{ORG} or a NE_{LOC}, and Rule TF3: if the first entity is a NE_{LOC}, keep the tuple if the second entity is also a NE_{LOC}; otherwise remove the tuple.

Thus, after applying filtering rules, the final set of entity tuples from the previous example is ⟨Barack Obama, Honolulu⟩ and ⟨Barack Obama, Columbia University⟩. Tuple ⟨Honolulu, Columbia University⟩ is filtered out due to Rule TF2, which is beneficial because a relation between Honolulu and Columbia University is not visibly helpful.

3.3 Relation Extraction

We denote the final set of entity tuples for a sentence after applying the filtering rules as F. Each tuple in F is in the format of *head-tail*, denoted as $\langle e_h, e_t \rangle$. Our algorithm for finding the relation between e_h and e_t is based on semantic matching.

Given a tuple $\langle e_h, e_t \rangle$, its sentence s and a set of pre-defined relations $R = (r_1, r_2, \ldots, r_n)$, we collect all the tokens between e_h and e_t in s (including e_h and e_t) and name this sequence of tokens as P_{sub}. For each relation r_i in R, we also construct a sequence of tokens as "$e_h r_i e_t$" and name it R_i. Using a state-of-the-art embedding model, Sentence-BERT (SBERT)[3] (Reimers and Gurevych, 2019), we compute the semantic similarity between P_{sub} and R_i by obtaining the embeddings of P_{sub} and R_i and computing their cosine similarity. We do this for all the r_i's in R and select the relation r_i whose R_i has the highest similarity score with P_{sub}. If this highest similarity score is higher than a threshold[4], then r_i is selected as the relation between e_h and e_t. This generates a triple (e_h, r_i, e_t) for the knowledge graph. This process is repeated for all the entity tuples for sentence s and for all sentences in the input document. A triple is removed if it has been generated from a previous sentence.

Figure 4 shows an example sentence, its two entities ⟨Barack Obama, Columbia University⟩, the P_{sub} formed by the two entities, the R_i's and the generated triple for the entity tuple. Note that even though the P_{sub} span is considerably long, SBERT helps generate the correct relation in this case because of contextual knowledge encoded within such pretrained language models, thus validating the effectiveness of using semantic similarity in

[2]https://spacy.io/

[3]We use *distilbert-base-nli-stsb-mean-tokens* as the pre-trained model.

[4]We set this threshold to 0.8 in our experiments based on the following experiment in the NYT dataset: beginning at 0 and increasing by 0.2 on each test until the threshold reaches 1, and we found that setting the threshold at 0.8 yielded the best F-score results. We use this threshold for all the other datasets.

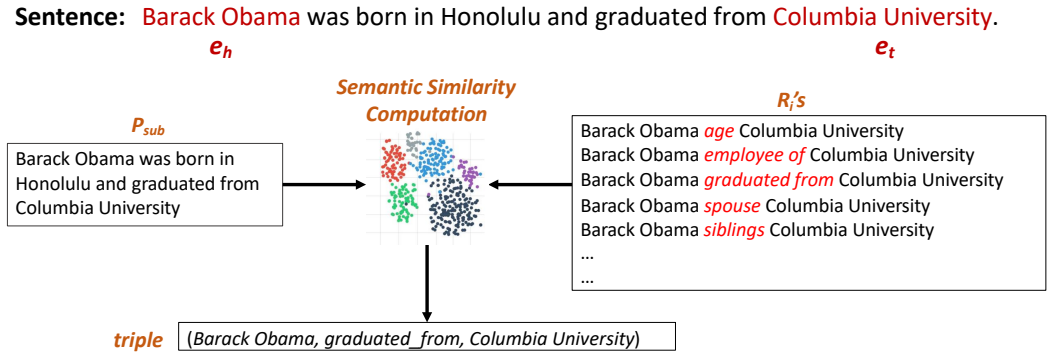

Figure 4: An example for Relation Extraction phase. At the top is the sentence with e_h and e_t denoting head and tail entities, respectively. P_{sub} is the part of sentence between e_h and e_t. R_i's are the sequences formed by the two entities and a relation. The final extracted triple for the two entities is also shown.

Dataset	# Sentence	# Relations	Example Sentences	Triple
TACRED	15509	42	*Both Konin and Alessi think so.*	(Alessi, no_relation, Konin)
TACRED*	3325	41	*Miettinen hired for WPS champ Sky Blue.*	(Miettinen, per:employee_of, Sky Blue)
NYT	5000	24	*At the time, she lived in Hollis, Queens.*	(Hollis, neighborhood_of, Queens); (Queens, contains, Hollis)
WEBNLG	703	246	*Bionico is a dessert containing sour cream from Mexico.*	(Bionico, country, Mexico); (Bionico, ingredient, cream)
NewsKG21	685	91	*Kevin Feige is married to Caitlin, a cardiothoracic nurse.*	(Kevin Feige, spouse, Caitlin); (Caitlin, job_title, cardiothoracic nurse)

Table 1: Dataset statistics. TACRED* is a subset of TACRED without instances containing triples with "no_relation".

KG relation extraction.

3.4 Optional Pattern-Based Rules

To further improve relation extraction in the news domain, we apply the following pattern-based rules based on our observation of their occurrence frequency in news articles: (1) Relation Extraction Rule 1 (RE1): if an entity tuple contains a noun phrase and a named entity of type Person (NE_{PER}) and the noun phrase is immediately before a NE_{PER} in the sentence (such as in "U.S. President Biden"), we assign "job title" as the relation; (2) Relation Extraction Rule 2 (RE2): if the two entities in a tuple appear as NE_{LOC}, NE_{LOC} in the sentence (such as in "Seattle, Washington"), the "is part of" relation is generated; and Relation Extraction Rule 3 (RE3): relation "job title" is generated in the tuple with the pattern $NE_{PER}, noun\ phrase$ (such as in "Caitlin, a cardiothoracic nurse").

We would like to emphasize that these rules are optional and even without these heuristics, our method outperforms the other unsupervised approaches, as demonstrated in Table 4 in section 5.3. Please also note that these rules may not be 100% accurate, but none of the existing KG generation methods is 100% accurate. These optional heuristics can better extract relations when two entities are next to each other in a sentence, where SBERT may not have enough information to correctly identify the relation between the two entities. We will show that these rules lead to a better overall result on news domains. Our goal here is to demonstrate that optional domain specific rules can be used to further improve the quality of the generated triples. If our purpose is to generate more labeled data for distant supervision, the use of these rules can reduce the overall noise ratio.

4 Evaluation Datasets

We evaluate our KG system by comparing the generated triples to manually annotated triples from three benchmark information extraction datasets and a new dataset on the news domain, all for English language.

4.1 Benchmark Datasets

The three benchmark datasets are: (i) **TACRED** (Zhang et al., 2017), (ii) **NYT** (Riedel et al., 2010), and (iii) **WEBNLG** (Gardent et al., 2017). Only their test datasets are used in our evaluation because our method does not need training. Each of the datasets contains a set of independent sentences and one or more ground truth triples for each sentence. TACRED has 41 relations originally from the TAC KBP yearly challenges [5] with a newly created relation called "no_relation"[6]. This dataset was manually constructed from an underlying corpus from TAC KBP where each sentence is labeled with a single ground truth triple and a standard evaluation tool is provided. NYT and WEBNLG datasets have 24 and 246 predefined relations, respectively. In both datasets, a sentence may have more than one ground truth triple. The statistics of the three benchmark datasets and our manually-created dataset are given in Table 1.

4.2 New Dataset: NewsKG21

Our goal in this research is to create a KG from news articles in order to build question-answering tools for editors of a news agency. The benchmark datasets we can obtain are not completely in the news domain. To evaluate our method on the news domain, we created a new dataset named `NewsKG21`. Another reason for us to develop a new KG generation dataset is that many public benchmark KG datasets are of poor quality since they were created mostly via crowdsourcing (e.g., in the TACRED dataset, the ground truth label for "AIG SELLS ALICO TO METLIFE" is ('ALICO', 'parents', 'AIG'), which is wrong). The evaluation results based on such datasets may be misleading. As a result, we carefully created a new dataset with as little noise as possible.

Four volunteers assisted in the creation of this dataset. One is an author of this paper, and the others are senior undergraduate Computer Science students. We selected 685 sentences from news articles published in 2021 in CNN, CBC, USNEWS, The Star, and Wikipedia News. From the 685 sentences, 1247 unique triples were manually generated. We divided the dataset into two parts: a test data set containing 271 sentences and 705 ground truth triples and a training set with 414 sentences and 542 ground truth triples. To prevent bias and advantages for a certain system, no system was engaged in the dataset creation process. Only the testing set is used to assess all unsupervised models.

5 Experiments and Discussion

5.1 Baselines and Metrics

We compare our system with two other state-of-the-art unsupervised systems[7], **Stanford OpenIE** (Angeli et al., 2015) and **MAMA** (with the BERT$_{LARGE}$ option) (Wang et al., 2020).

Entity tuple extraction: To compare the extracted entities with those in the ground truth data, we use *Token Set Ratio*[8], to calculate the similarity between two entities. Given an extracted entity E and the ground truth entity G, *Token Set Ratio* is defined as $\frac{2M}{T}$ where T is the total number of tokens in both E and G (that is, $|E|+|G|$ where $|X|$ is the number of tokens in entity X), M is the number of matched tokens between E and G, and tokens are separated by spaces in the entity (that is, tokens are basically the words in the entity). For example, if E is "Trudeau" and G is "Justin Trudeau", the token set ratio is 2/3.

This entity matching method is used for all the evaluated methods. Empirically, the threshold of string similarity is set to 0.9 for all the systems. The need for partial matching over exact matching is motivated by the observation that some gold standard annotations in the benchmark datasets are incompletely-matched entities. For example, "Apollo 12" appears as an entity in the original text, but it appears as "Apollo" in the gold standard triple in a benchmark dataset.

Triple generation: For a fair comparison, we also map the extracted relations from all the methods (including Stanford OpenIE and MAMA) to each of the dataset's relations using the same method, i.e., using SBERT embeddings for computing the cosine similarity between extracted relations and predefined relations in the schema, and selecting the one with the highest similarity score. We chose this relation mapping approach for Stanford OpenIE and MAMA instead of their original

[5] https://tac.nist.gov/

[6] The results of the evaluation including the "no_relation" instances can be found in Appendix A.

[7] Although Stanford OpenIE was trained in a semi-supervised way, we use their pre-trained version and do not fine-tune it on our training dataset. Thus, we consider our use of their method as unsupervised.

[8] https://pypi.org/project/fuzzywuzzy/

Dataset	System	P %	R %	F1 %
TACRED*	Stanford OpenIE	18.4	3.0	5.2
	MAMA	12.6	2.3	3.8
	(Ours) KGSS	**43.5**	**27.6**	**33.8**
NYT	Stanford OpenIE	2.7	1.5	1.9
	MAMA	1.7	7.2	2.8
	(Ours) KGSS	**25.7**	**29.2**	**27.3**
WEBNLG	Stanford OpenIE	2.5	6.5	3.6
	MAMA	5.1	6.0	5.5
	(Ours) KGSS	**8.4**	**9.1**	**8.7**
NewsKG21	Stanford OpenIE	7.1	11.3	8.7
	MAMA	2.1	6.1	3.2
	(Ours) KGSS	**24.6**	**20.4**	**22.3**

Table 2: The results of KG triple extraction.

System	P %	R %	F1 %
Stanford OpenIE	19.2	30.9	23.7
MAMA	11.4	32.8	16.9
KGSS	**45.1**	**48.7**	**46.8**

Table 3: Results of entity tuple extraction (e_h, e_t) on NewsKG21

System	P %	R %	F1 %
Stanford OpenIE	7.1	11.3	8.7
MAMA	2.1	6.1	3.2
KGSS (without rules)	**10.5**	**12.1**	**11.2**
KGSS with RE 1	**13.1**	**15.7**	**14.3**
KGSS with RE 1 & 2	**16.1**	**19.3**	**17.5**
KGSS with RE 1, 2 & 3	**16.5**	**20.1**	**18.1**
KGSS with 3 REs & tail type	**24.6**	**20.4**	**22.3**

Table 4: Results of triple extraction (e_h, r, e_t) on NewsKG21 dataset, without relation extraction rules (top) and with relation extraction rules (bottom). Adding rules improves the performance.

manual relation mapping techniques, which are irreproducible in our experiments.

For the TACRED* dataset, we calculate precision, recall, and F-score with the provided standard evaluation script. As the TACRED* dataset also contains pronouns and nouns as entities in the ground truth triples, we also extract these in addition to the named entities and omit the coreference resolution in our system for this dataset in order to have a fair comparison because both baselines can detect pronouns and nouns as entities. In our system, the user can choose types of entities that can be identified. For the NYT and WEBNLG datasets, we calculate the standard F1 score as $F1 = (2 * p * r)/(p + r)$, with $p = \frac{c}{m}$ and $r = \frac{c}{g}$, where c denotes the number of correctly extracted triples, m is the total number of extracted triples, and g is the number of triples in the annotated dataset.

5.2 Results and Discussion

Table 2 presents the results of KG triple generation over the four datasets. We note that our method KGSS consistently outperforms both unsupervised baselines across all the datasets by considerable margins on all the three metrics. One possible explanation for the improvement gains achieved by KGSS as compared to the unsupervised baselines is that the baseline methods tend to extract triples using verbs as signals which causes them to miss many triples, whereas our method generates the triples using semantic similarity from sentence embeddings. The baseline models also generate redundant triples which lowers their precision.

It is worth noting that among the four datasets, WEBNLG is the most challenging one for KGSS, with much lower performance than that on other datasets. This is most likely because of the large number of relation types in its schema (more than 200 as compared to other datasets having less than 100 relations). We conjecture that some relations may be too semantically similar for SBERT to distinguish from each other.

In terms of qualitative analysis, looking at the visual KG shown in Figure 3 generated for an excerpt from a Wikipedia article, we notice that all mentions of 'Bill Gates' and 'Gates' get correctly resolved to a single entity, i.e., 'Bill Gates', (and similarly, 'Microsoft Corporation' and 'Microsoft' get resolved to 'Microsoft Corporation') which helps prevent generating redundant triples. Another strength of the system can be seen in the form of triples such as ⟨Bill Gates, friend, Paul Allen⟩, ⟨Albuquerque, city of, New Mexico⟩ and ⟨Seattle, city of, Washington⟩. Also, all the various positions held by Gates are captured well, thus highlighting the role of such systems as helpful tools for summarizing long pieces of unstructured text into a concise visual representation.

5.3 Ablation Experiments

In Table 3, we evaluate the three systems on the NewsKG21 dataset one the task of entity tuple extraction, which means that we only compare the performance of systems generating pairs of head and tail entities to the ground truth in the dataset. We see that our method is better than Stanford OpenIE and MAMA which is most likely attributed to

our entity tuple filtering rules (TF1, 2, and 3) that can remove some noisy entity pairs while preserving a large number of meaningful tuples.

We also evaluate the three relation extraction rules described in Section 3.4. The results in Table 4 show that each rule helps to enhance the performance of our system as all the three measures increase as we apply more rules. The F-score is increased by around 7% after applying the three rules all together. One significant point to notice is that our system outperforms the other two unsupervised methods even when no heuristic rules are used.

By analyzing the generated triples, we realized that some incorrect triples can be avoided if we consider the entity types of a relation in relation extraction. For example, the spouse relation can only connect two entities of the person type. Thus, we add the type of the tail entity in each relation in our relation schema. Note the head entity type is already in the schema, similar to the schema in the TACRED dataset. With such information in the relation schema, we are able to eliminate some candidate relations given an entity tuple. For example, if the entity tuple is "Trump, New York", any relations whose head and tail entity types do not match Person and Location (such as the *spouse* relation) are not considered as candidates.

The last row of Table 4 demonstrates that by using the tail entity type for each relation in the schema, we can raise the F-score of our system by 4% points. This is another advantage of our system, which uses an entity-type aware method for eliminating unpromising triple extraction results, which the Stanford OpenIE and MAMA systems do not have. In addition, we run an ablation test on the NewsKG21 dataset using the tuple filtering criteria specified in section 3.2. As seen in Table 5, each rule contributes to the improvement of overall performance of our system.

One interesting finding is that, of the three systems, MAMA gets the lowest score on the NewsKG21 dataset since it extracts entity tuples based on information contained in a pre-trained language model BERT. As such, MAMA will approach its KG generation limit if the input articles are not from the language model's underlying corpus, such as our NewsKG21 dataset which is produced from the recent news stories.

Filtering Rule	P %	R %	F1 %
No Rule	12.9	25.4	17.1
TF 1	18.5	24.3	20.9
TF 1 & 2	19.1	23.4	21.1
TF 1, 2 & 3	20.9	23.4	**22.1**

Table 5: KGSS's performance on triple extraction with various tuple filtering methods on NewsKG21.

System	P %	R %	F1 %
Stanford OpenIE	33.5 ± 9.0	34.6 ± 15.9	34.0
MAMA	2.7 ± 2.6	10.3 ± 6.9	4.3
KGSS	34.1 ± 10.0	37.8 ± 12.7	**35.9**

Table 6: Results of human evaluation on the performance of triple extraction on NewsKG21.

5.4 Human Evaluation

In addition to automatic evaluation, we conduct human evaluation of our proposed system's triple extraction performance by comparing it to two baseline models: Stanford OpenIE and MAMA. Five human evaluators participated in our study, none of whom was told beforehand which systems they were assessing; more specifically, the names of each model were hidden. We chose 30 sentences at random from the NEWSKG21 dataset, and each participant graded the quality of triples generated by each system on each sentence based on the following criteria: (i) how accurate the extracted triples are in regard to the original text; and (ii) how thoroughly the extracted triples cover the true relations in the original sentence. Each evaluator was asked to assign a score from 0 to 1 to each generated triple on precision and to the set of triples generated from a sentence on recall, with 0 indicating entirely incorrect, 1 indicating completely accurate, and a value in between indicating partially correct.

The results in Table 6 show that Stanford OpenIE performs much better on human evaluation than on automatic evaluation. This is because only evaluating the system based on automatically match with the ground truth in the dataset may not accurately reflect the performance of a system. However, the results in Table 6 confirm that our system outperforms the two baseline models.

Although unsupervised approaches may allow more interpretable and flexible methods, they are not without limitations. The effectiveness of our unsupervised algorithm is partly dependent on the accuracy of the existing NER tools that we incor-

porate into our pipeline. Similarly, the semantic matching phase's performance may be less effective when the relation schema contains similar relation names. In addition, if training data are available, supervised methods can achieve much better results as shown in Table 9 in Appendix C. Nevertheless, our unsupervised method can work when no training data are available and can potentially be used to create labeled data (although noisy) for distant supervised learning to bootstrap knowledge graph generation.

6 Conclusions

We presented a novel unsupervised method for knowledge graph generation without the need for labeled data or manual mapping of extracted relations to a predefined relation schema (as in two previous unsupervised methods). A salient feature of the method is that it uses semantic similarity matching to find relations between entities. In addition, our system can work with any set of relations that the user prefers, flexibility that other methods, especially the supervised ones, do not have. We also created a new data set from news articles that will be shared with the community.

Our evaluation results demonstrate the effectiveness of our system which significantly outperforms two state-of-the-art unsupervised models over four different datasets. We also develop an open source interactive KG generation and visualization tool. As future work, we will evaluate effectiveness of using our method for bootstrapping knowledge graph generation with distant supervision.

Acknowledgements

We would like to thank our volunteer annotators Iris Chang, Rhitabrat Pokharel, and Andrew Jeon for their help in creating our NewsKG21 dataset. We are thankful to the anonymous reviewers for their helpful suggestions.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

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

Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Isaiah Onando Mulang', Saeedeh Shekarpour, Johannes Hoffart, and Manohar Kaul. 2021. RECON: relation extraction using knowledge graph context in a graph neural network. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1673–1685. ACM / IW3C2.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

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

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2006–2013. IOS Press.

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.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. 2020. Unsupervised relation extraction from language models using constrained cloze completion. *CoRR*, abs/2010.06804.

Wenti Huang, Yiyu Mao, Zhan Yang, Lei Zhu, and Jun Long. 2020. Relation classification via knowledge graph enhanced transformer encoder. *Knowledge-Based Systems*, 206:106321.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 105–113.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Jie Liu, Shaowei Chen, Bingquan Wang, Jiaxin Zhang, Na Li, and Tong Xu. 2021. Attention as relation: learning supervised multi-head self-attention for relation extraction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3787–3793.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 523–534.

Amit Singhal. 2012. Introducing the Knowledge Graph: things, not strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/. [Online; accessed 01-July-2021].

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7498–7505, Online. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Peilu Wang, Hao Jiang, Jingfang Xu, and Qi Zhang. 2019a. Knowledge graph construction and applications for web search and beyond. *Data Intelligence*, 1(4):333–349.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019b. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958.

Zina Wang. 2020. Unsupervised and supervised learning of complexrelation instances extraction in natural language. Master's thesis, Delft University of Technology.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

A Experiments on TACRED dataset including *no_relation* relationship

Table 7 compares our system's performance to Stanford OpenIE and MAMA on the TACRED dataset, which includes the relation: no_relation. In this experiment, if the relation confidence rate returned from SBERT is less than 0.8, our system will return no_relation. Although the total performance of all three systems decreases, our system still outperforms the other two cutting-edge models.

System	P %	R %	F1 %
Stanford OpenIE	6.6	3.0	4.1
MAMA	2.4	2.2	2.3
(Ours) KGSS	**14.3**	**27.6**	**18.8**

Table 7: The performance of triple extraction on TACRED including relationship "no_relation".

B Comparing performance of different algorithms on entity extraction

For entity extraction, we compare the performance of the named entity recognition (NER) systems

Dataset	Library	P %	R %	F1 %	Runtime (sec)
TACRED*	spaCy	29.3	86.4	43.7	145
	Stanza	29.2	88.6	43.9	1355
NYT	spaCy	57.5	99.6	72.9	51
	Stanza	56.8	99.9	72.4	454
WEBNLG	spaCy	86.7	86.5	86.6	6
	Stanza	91.5	91.6	91.5	47

Table 8: Performance of spaCy and Stanza for entity extraction

from two libraries, namely spaCy[9] and Stanza[10] (Qi et al., 2020) on the three benchmark datasets. A detected NE is considered to be correct if it partially matches the entities in the ground truth dataset via fuzzy string matching. The precision, recall, and F1 scores for both the tools are presented in Table 2, where we observe that while spaCy and Stanza are comparable in terms of their F1 scores, Stanza is about 8 times more computationally expensive. Thus, we select spaCy for NER and tokenization in all our experiments.

C Performance of the supervised KG models

Table 9 shows the performance of the state of the art supervised KG models: TransEN (Huang et al., 2020) on the TACRED dataset, and AaR (Liu et al., 2021) on the NYT and WEBNLG datasets. All the models are trained on the training data of each dataset and evaluated on the test data of the corresponding dataset. The results are taken from the references.

System	Dataset	P %	R %	F1 %
TransEN	TACRED	68.3	66.2	67.3
AaR	NYT	88.1	78.5	83.0
AaR	WEBNLG	89.5	86.0	87.7

Table 9: The performance of the state of the art supervised KG models on the TACRED, NYT, and WEBNLG datasets.

[9]https://spacy.io/

[10]https://stanfordnlp.github.io/stanza/