Unsupervised Relation Extraction with Sentence level Distributional Semantics

Manzoor Ali
DICE Group
Department of Computer Science
Paderborn University, Germany
manzoor@campus.uni-paderborn.de

Muhammad Saleem
DICE Group
Department of Computer Science
Paderborn University, Germany
saleem@informatik.uni-leipzig.de

Axel-Cyrille Ngonga Ngomo
DICE Group
Department of Computer Science
Paderborn University, Germany
axel.ngonga@upb.de

***Abstract*—Relation Extraction (RE) aims to identify the relationship between pairs of named entities in natural-language sentences. An unsupervised RE approach extracts relations in the absence of training data. Recently, many state-of-the-art unsupervised approaches have used word embeddings for RE. Such approaches ignore the semantic structure of the complete sentence. On the other hand, in this paper, we propose a novel approach that utilizes *sentence encoding* for unsupervised relation extraction. Our model classifies the sentence encoding of contextually similar natural-language sentences into clusters using an unsupervised approach, where each cluster consists of one or more potential relations. We queried the cluster for a candidate relation, and used a confidence value/threshold to extract accurate relations without semantic drift. We validated our approach by comparing it with both the unsupervised and bootstrapping approaches. Our experimental results suggest that our model achieves a better F-score on state-of-the-art datasets than the other unsupervised approaches.**

I. Introduction

Relation extraction plays an essential role in many natural language processing (NLP) applications such as knowledge base construction [1], event identification [2], and chatbot applications [3]. Researchers, particularly in the NLP community, have proposed various methods for relation extraction from natural language text. These methods include supervised, semi-supervised, unsupervised, and rule-based techniques [4]. Rule-based approaches extract relations with high precision, but they require human resources, domain knowledge, and a rule created for one type of natural language text may not apply to other kinds of texts. Supervised approaches achieve higher performance gains at the cost of creating quality-training labelled data [5]. Semi-supervised approaches have been employed to mitigate these requirements. Semi-supervised approaches(distant supervision and bootstrapping) suffer from incomplete knowledge bases-, and the availability of quality seeds [6]. The absence of labelled data or unavailability of quality seeds leads to unsupervised approaches. However, many of these approaches use hand-crafted features (e.g. dependency path and parts of speech) for relation extraction, which is time-consuming, complex, and often incomplete [7]–[9]. Recently, many state-of-the-art unsupervised approaches have used *word embedding* [10] for relation extraction using a combination of handcrafted features. These approaches consider word embedding for relation extraction, and ignore the semantic structure of the entire sentence. Consequently, the accuracy of word-embedding approaches decreases when the relationship depends on the context of other sentences. For example, *'...The main body of Golden Brown is in 6/8 time...'*, the word embedding approach extracts the *body_height(Golden Brown, 6/8)* relation. The actual relation in this sentence is *song_length(Golden Brown, 6/8)*. In such situations, sentence level distributional semantics (sentence encoding) performs better than word embedding in capturing the context of a sentence. We present a novel unsupervised relation extraction approach, called SURE, using SBERT [11] *sentence encoding* to overcome contextual limitations. We used SBERT instead of standard BERT-based encoding because it is computationally expensive [11]. Therefore, to the best of our understanding, this method has limited use in `unsupervised relational extraction`. We followed a transfer learning technique using pretrained state-of-the-art sentence encoding, SBERT [11]. This reduces the computational cost, which suits unsupervised relation extraction. Unlike other transfer learning approaches, the trained model was not fine-tuned. Instead, our model is built upon the common observation that sentence encoding captures contextual information better than word embedding [12]. We employed sentence encoding in a corpus of named entity-annotated sentences to generate vector representations. An unsupervised clustering algorithm combines similar vectors into clusters. Instead of considering all clusters, we chose only those clusters which are semantically close to a candidate relation. We introduce a query term approach, which is a natural language representation of a candidate relation, to extract the candidate relation. The system extracts semantically similar sentences to the query term. Both sentence level and word level distributional semantics often leads to semantic drift [13]. For example, *'the main company office in'* has a different meaning than *'a company branch office in'* but carries a high similarity score for headquarter relation. We calculate confidence scores to avoid semantic drift, and increases the precision of the relation extraction. Our main contributions are as follows:

- We utilize the sentence encoding for unsupervised relation extraction without any explicit feature selection.

- Our proposed algorithm achieves state-of-the-art results for unsupervised relation extraction.

The purpose of our relation extraction model is to identify all those sentences which contain a candidate (target) relation from a natural language text corpus. The formal definition and the details of our model are in section III. The source code, data, and instructions for reproducing the complete results are available from the GitHub repository https://github.com/manzoorali29/SURE.

II. Related Work

State-of-the-art in relation extraction can be subdivided into four categories. We summarize all of these in the beginning of this section. We then discuss the state-of-the-art unsupervised approaches.

Supervised methods. These approaches typically use linguistic features for relation extraction [4]. In general, they require a large amount of labelled data, which in most cases are unavailable or require too much human effort.

Distant supervision. These approaches use knowledge base entities, relations, and weakly labelled datasets based on heuristics for relation extraction [5], [14]. The advantage of distant supervision is that it requires a small amount of labelled data compared to fully supervised approaches. However, owing to the incompleteness of knowledge bases, such techniques may suffer from low accuracies [4].

Semi-supervised methods. In this approach, bootstrapping is used for relation extraction, based on the confidence score for a given relation [15], [16]. Bootstrapping requires quality seeds to extract relationships. However, finding quality seeds for each relationship is not always possible or sometime difficult [4], [6].

Unsupervised methods. These approaches extract relationships from unlabeled corpora. OpenIE [17] represents relations as unstructured text and uses phrases for relation extraction. OpenIE sometimes faces the problem of redundant extraction [18]; it considers different representations of the same relation as different relations. In addition to OpenIE, we further divided unsupervised relation extraction approaches into two categories:

1. Features-based approaches: We combined relation extraction approaches that use handcrafted features into this category. *RelLDA* and *RelLDA1* [19] follow a generative approach to extract relations; a sentence and an entity pair are considered a document, whereas the relation corresponds to a topic. RelLDA uses the shortest dependency path and the entity pairs. The RelLDA1 adds five features to RelLDA, including parts of speech and entity types. Simmon et al. [8] used a piecewise convolutional neural network (PCNN) approach for relation extraction. A state-of-the-art entity-type-based approach known as *EType* and *EType+* uses entity-type information to extract relations [9]. All of these approaches depend on different features and lag when a full context is required. Our proposed approach overcomes the feature-selection requirement and uses sentence encoding instead of word embedding. With sentence encoding, we achieved state-of-the-art results compared with these approaches.

2. Language models based approaches: Recent advancements in language models enable relation extraction systems to extract quality relations and achieve human-level performance on some QA (Question Answering) datasets [20]. We grouped all approaches that use language models for relation extraction into this category. Relation extraction uses the prediction capability of a language model to complete queries. Goswami et al. [21] extracted unsupervised relations from language models by using constrained cloze completion. Petroni et al. [22] associated factual evidence with a BERT-based model to improve answer generation. These language model approaches are relevant to our work because they use language models for unsupervised relational extraction. We compared our result with two state-of-the-art slot-filling relation extraction models, and our model achieved a higher F1 score. **Note** that our approach is the extended version of our work [12] based on the similar idea. Some key changes from our main idea are the comprehensive evaluation, formal representation, and changes in methodology.

III. Methodology

In this section, first we define our problem statement, followed by our methodology of solving it.

Definition III.1 (Problem Statement)**.** Let $\mathscr{S} := \{S_1, S_2..S_n\}$ be the set of all sentences from a corpus, where each sentence $S_i \in \mathscr{S}$ is represented as a triple $< \mathscr{E}, \mathscr{T}_E, T >$ with a set of entities $\mathscr{E} := \{E_1, E_2..E_n\}$, and a set of entity types $\mathscr{T}_E := \{\mathscr{T}_{E_1}, \mathscr{T}_{E_2}..\mathscr{T}_{E_n}\}$, and a sequence of tokens (words) T. A relation between two entities is represented by $R(E_h,\ E_t)$ where E_h and $E_t \in \mathscr{E}$, and the types of E_h and $E_t \in \mathscr{T}_E$. The goal is to find all sentences $\acute{S} \subseteq \mathscr{S}$ that contain the given relation $R(\mathscr{T}_{E_h},\ \mathscr{T}_{E_t})$, i.e., $\acute{S} := \{R \mid \exists S \in \mathscr{S} : R(E_h,\ E_t)\}$

For example, the sentence S_i: `Robinsons settled in Chicago` contain entities set $\mathscr{E} := \{$`Robinsons`, `Chicago`$\}$, and the entity types set $\mathscr{T}_E := \{$`PERSON`, `LOCATION`$\}$, and `place_lived` as a R between the two entities. The relationship $R(PERSON,\ LOCATION)$ in this sentence is represented as follows: $R(E_h,\ E_t) := place\_lived(Robinsons, Chicago)$

Fig. 1 shows the SURE architecture. Our model comprises four main modules: (1) Candidate sentences selection module that filters related sentences from corpus, (2) sentence encoding module SBERT [11], (3) clustering module, and (4) relation extraction.These components are discussed in details in the upcoming sections.

A. Candidate sentences selection

Our approach first filters the set of sentences $\mathscr{S}$ to obtain the set of candidate sentences C_R for the required relation $R(\mathscr{T}_{E_h},\ \mathscr{T}_{E_t})$ (Line 1 of Algorithm 1). This is performed by extracting *type* information for both entities (i.e., E_h, E_t) used in the given relation. These entity types are then compared against the entity types used in $\mathscr{S}$: all the sentences that contain both

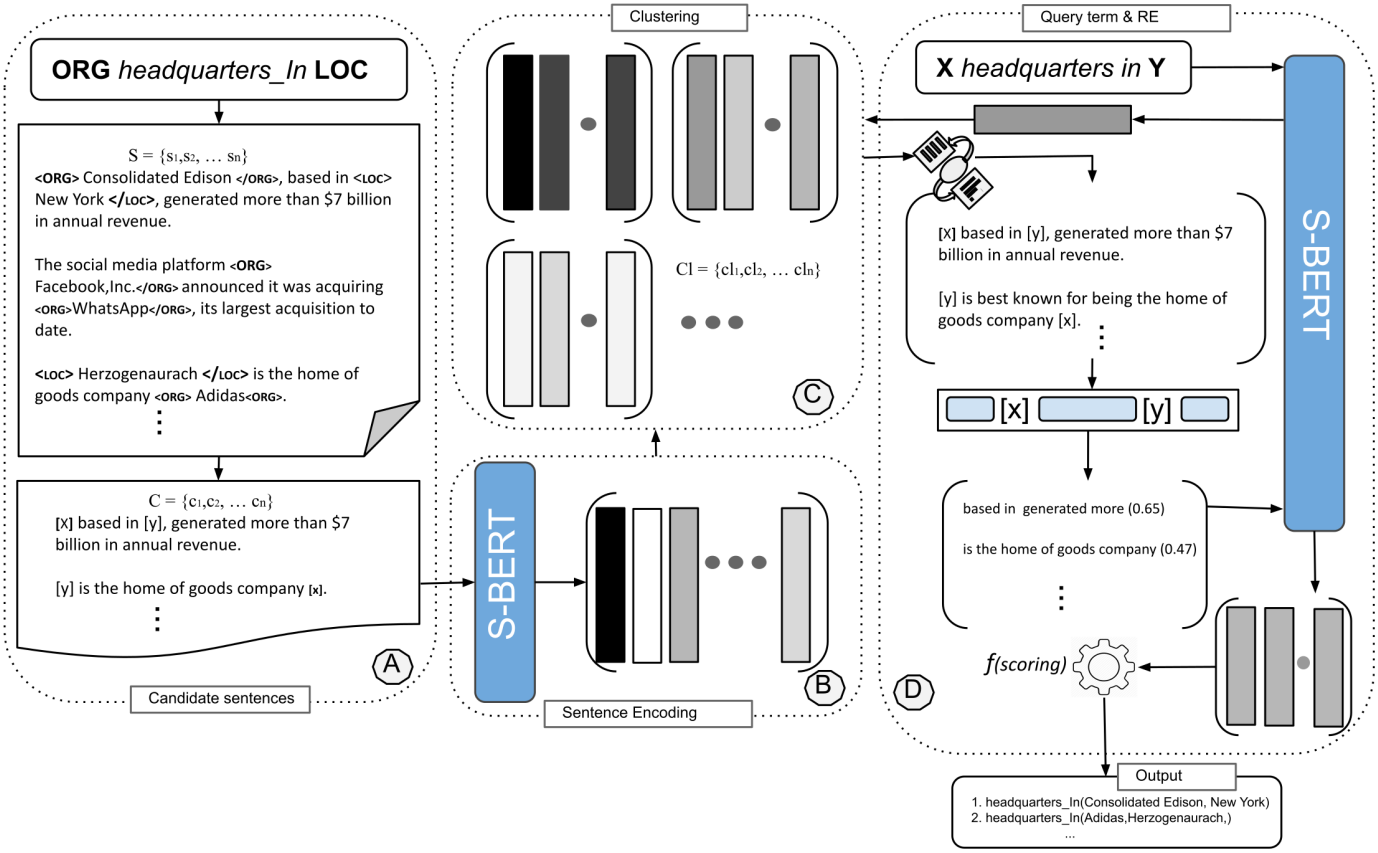

Fig. 1: SURE Architecture: A) Candidate sentences selection B) Sentence Encoding C) Clustering D) Query term encoding and relation extraction

entity types are selected into set C_R. Let $\mathscr{T}_{E_h}$ represent the *type* of the head entity E_h and $\mathscr{T}_{E_t}$ represent the *type* of the tail entity E_h of the relation, irrespective of the position of entities. Set C_R for the given relation is then defined as follows.

$$C_R := \{S \mid S \in \mathscr{S} : \mathscr{T}_{E_h} \in S \cap \mathscr{T}_{E_t} \in S\} \quad (1)$$

In Fig. 1, for a candidate relation `headquarters_in` with $\mathscr{T}_{E_h} :=$ organization (ORG) and $\mathscr{T}_{E_t} :=$ location (LOC), the model selects all sentences containing both organization and location entity types. For example, if a sentence contains ORG and LOC will be considered as candidate sentence, otherwise the sentence will be ignored for further processing. Our model is based on sentence context; regardless of the entity type position in a sentence. Therefore, a sentence is selected without considering the order of the entity types. Once the set C_R is selected, we replace all entities with entity types similar to E_h and E_t with common tokens, that is, E_h with [X] and E_t with [Y]. This substitution neutralizes the effects of entity strings in sentence encoding.

B. Sentence encoding

Owing to processing time complexity, traditional BERT-based [23] sentence encoding is generally not suitable for finding similarities between sentences from a large corpus [11]. For example, in finding the similarities between 10K sentences, BERT took approximately 65 h [11]. We used a pre-trained *distilbert-base-nli-stsb-mean-tokens* model of SBERT [11], which uses Siamese and triplet networks to encode sentences. Cosine similarity was used to compute the similarity among the encoded sentences. In addition, SBERT applies MEAN pooling to produce fixed-size vector sets $\mathscr{H} := \{H_1, \ldots, H_n\}$ (Line 2 of Algorithm 1), where

$$H_i := SBERT(MEAN\_Pooling(BERT(S_i))) \quad (2)$$

The pre-trained model was trained on 570K sentences. This model produces a 768-multi-dimensional vector for each encoded sentence.

C. Clustering

This step aims to classify similar vectors (i.e., semantically similar sentences) into clusters (Line 3 of Algorithm 1). Each cluster can contain one or more potential unlabeled relations. Unsupervised clustering algorithms such as K-means and K-medoids require the number of clusters to be determined or estimated before running the algorithm. Therefore, we chose an unsupervised version of adaptive affinity propagation that automatically selects its centroid and number of clusters from the corpus. Grouping similar sentences into clusters reduces the computational cost by only searching the relevant cluster for a given relation instead of considering all of them. We only

Algorithm 1: SURE Relation Extraction

Require: $\mathscr{S} := \{S_1, \ldots, S_n\}$,$R < \mathscr{T}_{E_h}, \mathscr{T}_{E_t} >$
Ensure: $\acute{S} \subseteq \mathscr{S} := \{S_1, \ldots, S_m\}$
1: $C_R := candidateSentences(\mathscr{S})$
2: $\mathscr{H} := SBERT(C_R)$
3: $\mathscr{C} := getClusters(\mathscr{H})$
4: $H_q := queryTerm(R)$
5: **for each** $C \in \mathscr{C}$ **do**
6: $Ex := getExemplars(C)$
7: $sim := COS(Ex, H_q)$
8: **if** $sim > \theta$ **then**
9: $\acute{\mathscr{C}} := \acute{\mathscr{C}} + C$
10: **end if**
11: **end for**
12: **for each** $\acute{C} \in \acute{\mathscr{C}}$ **do**
13: $[topK] := selectTop(sim)$
14: $\acute{S} := \acute{S} + sentence([topK])$
// call second iteration
15: $\acute{S} := \acute{S} + outPut(\acute{C}, [topK], sim_p)$
16: **end for**
17: **return** $\acute{S}$

Function outPut (H_C, H_q, sim_p) **:**
18: **for each** $H_c \in H_C$ **do**
19: $sim_c := COS(H_c, H_q)$
20: $Pscore := getPScore(sim_p, sim_c)$
21: $H_t := getTokens(sentences(H_c))$
22: $PMI := getPMI(H_t, H_q)$
23: **if** $Pscore$ and $PMI > 0$ **then**
24: $\acute{S} := \acute{S} + sentence(H_c)$
25: **end if**
26: **end for**
return $\acute{S}$

perform clustering when the entity types for a given relationship change. For example, for relations such as birthPlace, livedIn, studiedAt, we perform clustering once because the entity types remain the same: PERSON and LOCATION.

D. Query encoding and relation extraction

In this step, we first generate the natural language representation of the given candidate relation $R(\mathscr{T}_{E_h}, \mathscr{T}_{E_t})$. We can manually create these representations for candidate relations using the RE-Flex [21] approach, where they create cloze template (natural language representation) for each relation manually. For example, for the relationship headquarter, we can choose *headquarters in*. In our experiments, we extracted these representations from the item labels of the Wikidata RDF dataset.

We then create the query term by appending the neutralization tokens [X] at the beginning and [Y] at the end of the natural language representation of the candidate relation. In our motivating example, the *query term* is the, `[X] headquarters in [Y].` We also used sentence encoding (Section III-B) to transform the query term into its vector representation H_q (Line 4 of Algorithm 1). We now compute the cosine similarity between the query term H_q and exemplars Ex from each cluster (lines 5- 7 of Algorithm 1). All exemplars, that have a similarity value above a given threshold, the corresponding clusters, are selected for further processing (Lines 8 - 9 of Algorithm 1). For each selected cluster, we chose top-k vectors (i.e., sentences) based on the high cosine similarity score (Line 13 of Algorithm 1). This top-k sentences are added into the finally selected sentences set $\acute{S}$ (Line 14 of Algorithm 1). At this stage, we assume that these top k sentences have a high precision (later our results confirm that); however, we further need to increase the recall. For this purpose, we represent each sentence in the selected top-k sentences as a new query term and recompute the cosine similarity with the remaining vectors within the same cluster. All the vectors which have similarity score higher than the predefined threshold are also added into the set $\acute{S}$ (Lines 18-26 of Algorithm 1). Note that every selected vector H_c in the second iteration has a parent vector H_{pc} in the first iteration, that is, H_{pc} was used as an input query term which resulted in the selection of H_c. This second iteration increases the recall; however, it may result in semantic drift owing to multiple query terms (different from the original query term) selected in the second iteration. To overcome the effect of semantic drift, we defined a *Pscore* as follows:

$$P_{score} := COS(H_c, H_{pc})^2 + COS(H_{pc}, H_q) - 1 \tag{3}$$

where H_q is a vector representation of the original query term. In the second iteration, we consider only those sentences for which the $P_{score} > 0$ (Line 23 of Algorithm 1).

Furthermore, the probability of correct relation extraction increases if the relational phrase (i.e., query term) exists near the head and tail entities annotated in the sentences [15]. To increase the probability of a possible relation between two entities, we use a windows-based approach such as snowball [15] by selecting tokens (i.e. before, between, and after entities) around the head and tail entities. We calculate the pointwise mutual information (PMI) using the vector representation of the query term and vector representation of the selected window [24] (i.e., instead of the complete sentence) as follows.

$$PMI := \frac{< H_q, H_w >}{||H_q||\,||H_w||} \tag{4}$$

where H_q and H_w represent the vector representation of the query term and the selected window part of the sentence, respectively. To increase precision, we only consider a sentence as a relational sentence; if the PMI score is higher than zero, otherwise, it is ignored (Line 23 of Algorithm 1).

IV. Experimental Setup

A. Datasets

We used three standard relation extraction datasets to evaluate our model: (1) The New York Times (NYT-FB) FreeBase dataset [25] containing 455,771 training sentences and 172,448 test sentences. The dataset has 53 relations, including NA (no relation). Many state-of-the-art unsupervised relation extraction models have used this dataset for relation extraction evaluations [8], [9], [19]. (2) The Wikipedia-Wikidata-relations dataset [26][1], comprises Wikipedia sentences aligned with Wikidata relations. This dataset comprises the training, testing, and evaluation sets. The training set contained 372,059 sentences, while the testing set contained 360,334 sentences. Sentences in this dataset can have multiple relationships. The dataset contains a total of 353 unique relations, which have also been used in recent evaluations [27], [28]. (3) The English Gigaword dataset [29] is the third dataset used to compare our model with the bootstrapping-based approach. We selected one million already annotated sentences from the English Gigaword dataset. We chose this dataset because it does not have any labels; therefore, it is suitable for unsupervised RE. It is used in many state-of-the-art evaluations [16], [30]. We evaluate our model based on the method proposed by Bronzi et al. [31]. For the database, we used the DBpedia [32] and Freebase Easy dataset [33]. We created a combined (DBpedia and, Freebase) set of entities and relations for the four selected relations.

B. Selected models for comparison

We select relevant models for evaluation whose implementation is publicly available.

Unsupervised models: For unsupervised models, we selected RelLDA1 [19] as a generative approach, and March [7], Simon [8], and EType+ [9] as three feature-selection-based models for relation extraction. They used the NYT-FB dataset in their evaluations; therefore, we compared them with our model using the same dataset. We used an open-source implementation of these models on NYT-FB dataset, all these models also used NYT-FB for evaluation.

Language-model-based systems: In this category, we selected RE-Flex [21] and GD [22], which are two state-of-the-art QA models. RE-Flex uses inferences from language model predictions to answer queries. GD concatenates the context with a question to predict a correct answer from the language model. We used the Wikidata dataset to compare models in this category because, unlike NYT-FB, all sentences in the Wikidata dataset were annotated and contained relations. We converted the Wikidata dataset to the RE-Flex proposed architecture, where we extracted the subject and object from a sentence and predicted their relation. The motivation behind selecting these models is the unsupervised approach and use of language models. These approaches use a query to complete the relation, which is another aspect similar to our model.

Bootstrapping models: We selected the well-known BREDS [16] system, a semi-supervised bootstrapping approach for relation extraction that uses word embedding. We adopted the evaluation methods used in the BREDS to compare our results. We evaluate our system and BREDS on the English Gigaword dataset, [29] mentioned in the BREAD [16] paper.

C. Metrics

For the Wikidata dataset, we used both micro and macro precision, recall, and F-score [26]. For the NYT-FB dataset, we used the micro-precision, recall, and F-score according to [34], that is, we used P@10 and P@30 for precision and recall, and then took the average of the precision and recall to calculate the F-score. On the English Gigaword dataset, we used macro scores only for precision and recall because we needed to evaluate our model on predefine relations given by BREDS [16]. We used a precision metric for manual evaluation in our ablation study by computing the inter-rater agreement (the degree of agreement among raters). We evaluated our model based on the final output; therefore, we ignored other metrics related to cluster quality, such as B^3, V, and ARI.

D. SURE configuration

We configured the SURE model by applying various contextual input vectors as below:

SURE (without neutralization): This implementation uses the sentences without replacing the head and tail entity strings with [X] and, [Y], respectively. Here, we skipped the windows-based approach, that is, we kept all the sentence tokens.

SURE (without PMI): This implementation uses a window-based approach without considering the additional filter imposed by PMI. This way, we consider tokens around the entities as a complete sentence while ignoring the calculation of PMI score.

SURE: This implementation is a complete model according to section III.

Hyperparameters In our final evaluation, we set θ (a threshold value) equal to 0.35 according to Fig. 2. We achieved the best recall, precision and F-score for this value based on an empirical assessment performed with different θ values. The F-score on 0.35 changes slightly, but we kept a balance in all the three parameters (recall, precision and F-score). Note that user can adjust this value according to the requirement of precision or recall. We used the following window sizes during our experiments: i) *Before:* We selected two tokens before the first entity. ii) *Between:* eight tokens between the two mentioned entities; and iii) *After:* two tokens after the second entity. These are configurable values that can be adjusted according to the required precision [2].

[1]We name Wikipedia-Wikidata-relations dataset as Wikidata dataset in the rest of our paper.

[2]Further details about the hardware and software requirements are available on GitHub.

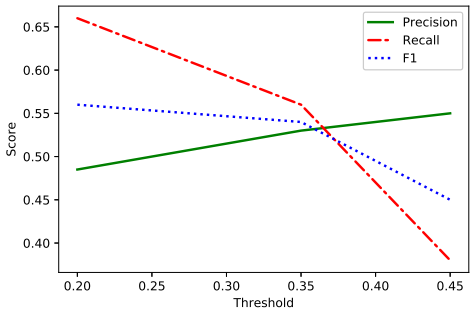

Fig. 2: Precision, recall and F1 scores on different threshold

TABLE I: SURE comparison with unsupervised models on NYT dataset using two NER systems(Stanford NER and AllenNLP NER)

	Stanford			AllenNLP		
Models	P	R	F1	P	R	F1
RelLDA1	0.31	0.46	0.37	-	-	-
March	0.31	0.51	0.38	0.32	0.52	0.40
Simon	0.33	0.51	0.40	0.33	0.50	0.40
EType+	0.30	**0.61**	0.40	0.31	**0.64**	0.42
SURE (Without PMI)	0.37	0.47	0.41	0.38	0.61	0.47
SURE	**0.41**	0.46	**0.43**	**0.41**	0.60	**0.49**

V. Results

A. Results on NYT-FB dataset

Originally, the NYT-FB dataset was annotated using StanfordNER. We ran the AllenNLP NER on the dataset to annotate the sentences and performed our experiments on both annotators. Table I presents the evaluation results for the NYT-FB dataset. As an overall evaluation of the F1 scores, we surpassed the state-of-the-art by up to 8%. In general, the selected models performed better on AllenNLP NER than StanfordNER because the accuracy of the AllenNLP NER annotation was generally higher than that of Stanford NER [35]; therefore, it helped the models to extract more accurate relations. Table I shows that our precision, in general, is higher than that of the other models because of the strict filters on the θ, and PMI values used. However, this may have resulted in a decrease in the recall.

B. Results on Wikidata dataset

Table II shows a comparison of our approach with language-model-based relation extraction models. We outperformed the selected models by more than 9% in the micro F1 scores. We obtained a higher F1 score on this dataset than on the NYT-FB dataset. The F1 score increased for the Wikidata dataset compared with the NYT-FB. The reason for this increase is the number of annotations in both the datasets. The entities in the Wikidata dataset were annotated in almost 100% of the sentences. In contrast, NYT-FB have approximately 35% of the sentences annotated.

We also observed that some sentences that appear in two or more relations decrease our model accuracy on the Wikidata dataset, one reason for these results is a potential relation that is not explicitly indicated. This also occurs because of the Wikidata dataset structure, primarily created for two or more relations in one sentence. Overall, our model performs better on the Wikidata dataset than on the NYT-FB dataset because the Wikidata dataset is appropriately aligned to the Wikidata entities. The macro score was relatively low owing to the relationship imbalance in the Wikidata dataset.

TABLE II: Comparison with language-model-based systems: Precision (P) Recall (R) and F1 score on Wikidata dataset.

	Micro			Macro		
Models	P	R	F1	P	R	F1
GD	0.34	0.22	0.27	0.25	0.11	0.15
RE-Flex	0.44	0.56	0.49	0.34	0.14	0.20
SURE(Without PMI)	0.45	0.53	0.48	0.31	0.16	0.21
SURE	**0.47**	**0.84**	**0.58**	**0.36**	**0.16**	**0.22**

TABLE III: Precision, Recall and F1 score for selected relations with bootstrapping based Approach (BREDS) using the English Gigaword dataset

	BREDS			SURE		
Relations	P	R	F1	P	R	F1
birthPlace	0.45	0.77	0.57	0.55	0.75	**0.63**
locatedIn	0.51	0.79	0.62	0.70	0.73	**0.71**
headquarters	0.66	0.70	**0.68**	0.59	0.64	0.61
founderOf	0.88	0.84	**0.86**	0.91	0.80	0.85

C. Results on English Gigaword dataset

Table III compares our model with BREDS for the four selected relations (used in the evaluation performed in the BREDS paper). Our model achieved a higher precision score for three out of four relationships. In terms of recall, BREDS had higher values than our approach. As mentioned before, because of the strict filters used in the candidate sentences selection, our precision is generally high, and recall is slightly low. The average F1 score for the four relations of SURE was higher than BREDS by 2 %. The results show that SURE can outperform bootstrapping-based approaches without any seed information.

VI. Ablation Study

Effectiveness of Windows-based token selection: We compared Windows-based performance with complete sentence. The results based on the NYT-FB dataset are shown in Fig 3. It can be observed that Windows-based tokens selection increased the precision (0.57 vs. 0.62) and decreased the recall (0.77 vs. 0.79). However, the overall F1 score was increased by selecting the Windows-based sentence. The Precision is increased because the nearby tokens (selected by the Windows-based approach) generally contain more semantic information about the relation. Therefore, they lead to more accurate relation prediction. We also observed that the Windows-based

TABLE IV: Extracted patterns in two iterations with similarity score to the selected query terms

Relations	Query terms	First Iteration		Second Iteration	
		Top patterns	**Score**	**patterns**	**Score**
birthPlace	born in	a native of,	0.81	<E >of <E >	0.57
		who grew up in,	0.85	and others native of	0.67
		who is originally from <E >and now,	0.68	<E >'s doorstep in Harvard Square in	0.47
locatedIn	located in	for the facility in,	0.88	mine in <E >who owns a house in <E >	0.54
		the capital of <E >the,	0.74	the trendy center of <E >	0.71
		in the capital of where,	0.70	-	-
headquarter	headquarters in	have the headquarters in <E >that,	0.79	building in	0.76
		head office in,	0.80	<E >researchers , who are based in	0.69
		visit the <E >facilities in <E >near,	0.56	player in <E >who originally committed to <E >	0.38
		a company based in'	0.82	but has executive offices in	0.79

approach was beneficial for relation prediction in the long sentences.

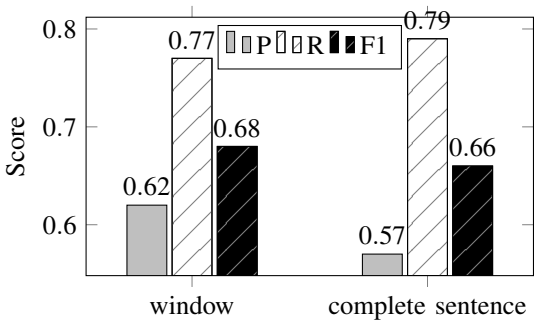

Fig. 3: The impact of selection of window-based approach vs complete sentences on the F1 score

Value added by the PMI score: We studied the effect of the PMI score for the top six relations of the Wikidata dataset. Table V suggests that the PMI score reduces the recall but improves the precision and F1 score for all six relationships. RE-Flex reported similar results.

The impact of neutralization: We include neutralization by replacing the entity text in a sentence with two common tokens, [X] and [Y]. The query term also contained tokens [X] and [Y]. Fig 4 shows the impact of neutralization. The recall remains the same, although there is a clear difference in precision. Sentence neutralization also helps in parameter setting because it increases the similarity score.

TABLE V: Precision (P) Recall (R) score for top Wikidata relations, using different configuration of SURE

	SURE(Without PMI)		**SURE**	
Relations	P	R	P	R
Shares border with	0.44	0.94	0.47	0.79
Instance of	0.22	0.96	0.31	0.62
Citizenship	0.61	0.97	0.62	0.89
Subclass of	0.57	0.67	0.37	0.45
Located in	0.56	0.74	0.57	0.41
Part of	0.45	0.78	0.54	0.39

TABLE VI: Manual evaluation of randomly selected facts

Relations	True(T)	False(F)	Ambiguous(A)	Precision
birthPlace	66	6	3	0.88
locatedIn	62	8	5	0.83
headquarters	65	8	2	0.87
founderOf	72	3	0	0.96

Similarity scores in the first and second iterations : Table IV lists the cosine similarity values obtained for different patterns in the first and second iterations of our relation extraction algorithm. We observe that the similarity scores in the first iteration are higher than those in the second iteration. This is because we used multiple query terms in the second iteration instead of a single original query term as in the first iteration. The low similarity also suggests a gradual shift towards semantic drift. Recall that we used the second iteration to increase the overall recall of our score. For example, for relation *headquarter*, the pattern *has an executive office in* was missed in the first iteration but captured in the second iteration.

Manual evaluation In addition to our automatic evaluation, we also evaluated our model using four human volunteers. We selected 75 randomly extracted assertions for each of the top four relations of the English Gigaword dataset. It was

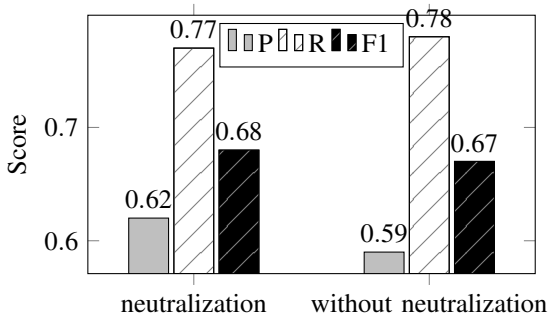

Fig. 4: The impact of neutralization vs entity string

impossible to calculate the recall, and hence, the F1 score based on human evaluation. Thus, we only report the precision values. We also computed the inter-rater agreement for each fact. If three out of four volunteers agreed on the assertion having been extracted correctly, we considered the assertion to be true. Otherwise, it was considered false. The results of the 300 randomly selected facts evaluated by humans are presented in Table VI. The results suggest that our approach achieved even better precision scores than those reported by benchmarks.

VII. Conclusion

This paper presents SURE, an unsupervised approach that captures semantic information from sentence encoding for relation extraction. Our empirical study suggests that sentence encoding can improve unsupervised relation extraction. From our evaluation, we can conclude that: (1) our approach performs better than state-of-the-art unsupervised relation extraction models without the need for any explicit feature selection (e.g., parts of speech, dependency tree, surface form), (2) a single query term for a particular relation can extract quality relations without semantic drift, (3) Windows-based token selection and PMI increases the F1 score, and (4) capturing the semantics of the overall sentence helps in predicting more accurate relations.

References

[1] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 455–465.

[2] Q. Ning, Z. Feng, and D. Roth, "A structured learning approach to temporal relation extraction," *arXiv preprint arXiv:1906.04943*, 2019.

[3] M. Yu, W. Yin, K. S. Hasan, C. d. Santos, B. Xiang, and B. Zhou, "Improved neural relation detection for knowledge base question answering," *arXiv preprint arXiv:1704.06194*, 2017.

[4] S. Pawar, G. K. Palshikar, and P. Bhattacharyya, "Relation extraction: A survey," *arXiv preprint arXiv:1712.05191*, 2017.

[5] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of ACL*, 2009, pp. 1003–1011.

[6] A. Smirnova and P. Cudré-Mauroux, "Relation extraction using distant supervision: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–35, 2018.

[7] D. Marcheggiani and I. Titov, "Discrete-state variational autoencoders for joint discovery and factorization of relations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 231–244, 2016.

[8] É. Simon, V. Guigue, and B. Piwowarski, "Unsupervised information extraction: regularizing discriminative approaches with relation distribution losses," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1378–1387.

[9] T. T. Tran, P. Le, and S. Ananiadou, "Revisiting unsupervised relation extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7498–7505. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.669

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[11] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[12] M. Ali, M. Saleem, and A.-C. N. Ngomo, "Unsupervised relation extraction using sentence encoding," in *The Semantic Web: ESWC 2021 Satellite Events*. Cham: Springer International Publishing, 2021, pp. 136–140.

[13] A. Blank, "Why do new meanings occur? a cognitive typology of the motivations for lexical semantic change," *Historical semantics and cognition*, vol. 13, no. 6, 1999.

[14] R. Bunescu and R. Mooney, "Learning to extract relations from the web using minimal supervision," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 576–583.

[15] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *ACM conference on Digital libraries*, 2000, pp. 85–94.

[16] D. S. Batista, B. Martins, and M. J. Silva, "Semi-supervised bootstrapping of relationship extractors with distributional semantics," in *In Empirical Methods in Natural Language Processing*. ACL, 2015.

[17] A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007, pp. 25–26.

[18] C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh, "A survey on open information extraction," *arXiv preprint arXiv:1806.05599*, 2018.

[19] L. Yao, A. Haghighi, S. Riedel, and A. McCallum, "Structured relation discovery using generative models," in *proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, pp. 1456–1466.

[20] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[21] A. Goswami, A. Bhat, H. Ohana, and T. Rekatsinas, "Unsupervised relation extraction from language models using constrained cloze completion," *arXiv preprint arXiv:2010.06804*, 2020.

[22] F. Petroni, P. Lewis, A. Piktus, T. Rocktäschel, Y. Wu, A. H. Miller, and S. Riedel, "How context affects language models' factual predictions," *arXiv preprint arXiv:2005.04611*, 2020.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[24] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "A latent variable model approach to pmi-based word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 385–399, 2016.

[25] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

[26] D. Sorokin and I. Gurevych, "Context-aware representations for knowledge base relation extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1784–1789.

[27] C. Christodoulopoulos and A. Mittal, "Simple large-scale relation extraction from unstructured text," *arXiv preprint arXiv:1803.09091*, 2018.

[28] A. Bastos, A. Nadgeri, K. Singh, I. O. Mulang, S. Shekarpour, and J. Hoffart, "Recon: Relation extraction using knowledge graph context in a graph neural network," *arXiv preprint arXiv:2009.08694*, 2020.

[29] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword," 2011.

[30] D. Yu, K. Sun, C. Cardie, and D. Yu, "Dialogue-based relation extraction," *arXiv preprint arXiv:2004.08056*, 2020.

[31] M. Bronzi, Z. Guo, F. Mesquita, D. Barbosa, and P. Merialdo, "Automatic evaluation of relation extraction systems on large-scale," in *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, 2012, pp. 19–24.

[32] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[33] H. Bast, F. Bäurle, B. Buchhold, and E. Haußmann, "Easy access to the freebase dataset," in *Proceedings of World Wide Web Conference*, 2014, pp. 95–98.

[34] P. Xu and D. Barbosa, "Connecting language and knowledge with heterogeneous representations for neural relation extraction," *arXiv preprint arXiv:1903.10126*, 2019.

[35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.