Impact of Entity Graphs on Extracting Semantic Relations

Rashedur Rahman, Brigitte Grau, Sophie Rosset

Impact of Entity Graphs on Extracting Semantic Relations

Rashedur Rahman[1], Brigitte Grau[2], and Sophie Rosset[3]

[1] IRT SystemX, LIMSI, CNRS, Université Paris-Saclay, France
rashedur.rahman@irt-systemx.fr
[2] LIMSI, CNRS, ENSIIE, Université Paris-Saclay, France
brigitte.grau@limsi.fr
[3] LIMSI, CNRS, Université Paris-Saclay, France
sophie.rosset@limsi.fr

Abstract. Relation extraction (RE) between a pair of entity mentions from text is an important and challenging task specially for open domain relations. Generally, relations are extracted based on the lexical and syntactical information at the sentence level. However, global information about known entities has not been explored yet for RE task. In this paper, we propose to extract a graph of entities from the overall corpus and to compute features on this graph that are able to capture some evidences of holding relationships between a pair of entities. The proposed features boost the RE performance significantly when these are combined with some linguistic features.

1 Introduction

Relation extraction (RE) from text is a useful task for populating Knowledge Base about entities. Many relations exist between pairs of entities and RE systems learn how the relations between entity-mentions are expressed in texts. RE systems make use of linguistic features based on semantic [1] and syntactic analysis [2, 3]. Recently neural networks have been applied for RE task that use word embeddings for semantics without requiring complex feature engineering [4]. These methods use local information at the sentence level but do not account for global information on the entities at collection level. Recent work on Web RE [5] studied global information about the object entity and words around the entity-mentions. Such information facilitates introducing some sort of world knowledge for making choices in addition to modeling the linguistic expression of relation in sentences.

We hypothesize that a pair of entities (subject and object) having a true relationship should share more common neighbors than a false relationship between that particular subject and a different object. For example, the spouse of a person should share more places and relationships with his/her spouse than with a person who has no true relation with him/her. Therefore, we construct a graph of entities based on a corpus that allows us to propose new characterizations of the relations by community graph-based features [6, 7], in addition to newly defined linguistic features. In [8], we have shown the effectiveness of graph based features for validating claimed relations on a relatively small dataset and a large number of relations. Thus we go further in our study in order

to enlarge the training and test data and closely observe the impact of graph features on extracting some semantic relations.

For evaluating the relevance of the proposed features, we tested them on a task of relation validation (RV). RV examines the correctness of relations that are extracted by different RE systems. It facilitates to evaluate the new features without developing a complete RE system. In this paper, we add a new kind of evaluation and we evaluate the proposed features on knowledge base population (KBP) where the validation results are used for choosing entities that fill relations linked to a query entity. Experimental results show that the newly proposed features lead to outperform the RV baseline by around 10 points. The KBP evaluation also shows improvement over state-of-the-art system results by 1.5 point.

2 Related Works

Several features have been explored for relation extraction (RE) from texts. Existing RE methods basically extract linguistic evidences of holding relationship between two objects at the text level based on syntactic and semantic analysis.

Syntactic analysis captures the grammatical structures of expressing relations among different words in a sentence. Therefore, syntactic dependency has been widely explored for RE task [9, 10]. In [2], dependency tree has been used for defining kernel functions based on the shortest dependency path between two entities. Sometimes, shortest path fails to capture enough information for RE therefore, a context-sensitive convolution tree kernel [9] was proposed to include necessary information outside the shortest path. In open information extraction [11], dependency parsing was employed to define some patterns of relations and to discover verb-clauses at sentence level.

However, syntactic information cannot characterize the semantic type of a relation. Therefore, lexical features i.e. words between and around the mentions are effective [1, 12–14] for RE task. Dependency trees and trigger words were combined to take advantage of both syntactic and semantic features for bio-medical RE in [15].

Dependency and lexical features have been used in the existing rule based [3, 11] and supervised [1, 9] methods of RE. Some feature based RE methods [14, 16, 17] used POS-tags in addition to the dependency and word features. Rule based methods are restricted to extract a small number of relations while supervised methods are very effective but require a large amount of labeled data. Distant supervision [12, 16, 17] does not require manually labeled data for learning relations. These methods inherit state-of-the-art linguistic features and apply some probabilistic models for extracting relations. Nowadays, patterns and semantic types of relations are learned automatically by employing word embeddings and neural networks [18–20].

However, no existing method used entity level global information for RE task. Some kind of global information about the object entities has been studied in Web RE task [5]. Global information about entities gives some clues how the entities are associated among them. Such information can be explored by representing entities as nodes in a graph.

A graph structure facilitates analyzing paths between nodes and relationships among them. Several graph based methods have been proposed for different tasks i.e. automat-

ically completing existing knowledge base [21, 22], automatic trigger identification for slot filling [23], entity linking [6, 24] etc.

Several features have been computed on graph i.e. entropy for discovering knowledge in publication networks [25], centrality measurements for finding important and influential nodes in social networks [7] etc.

We construct a graph of entities after extracting named entities from a collection of texts and we propose some new features for RE which are computed on the graph of entities by analyzing the communities of pair of entities.

3 Community Graph of Entities

3.1 Definition of the Graph

Let a graph $G = (E, R)$, a query relation (slot) r_q, a query entity $e_q \epsilon E$, candidate responses $E_c = \{e_{c1}, e_{c2}, \ldots, e_{cn}\} \epsilon E$ where $r_q = r(e_q, e_c) \epsilon R$. The list of candidates is generated by different relation extraction systems. Suppose other relations $r_o \epsilon R$ where $r_o \neq r_q$. We characterize whether a candidate-entity e_{ci} of E_c is correct or not for a query relation (r_q) by analyzing the communities X_q and X_c formed by the query entity and each candidate response. A community X_i contains the neighbors of e_i, and this up to several possible steps.

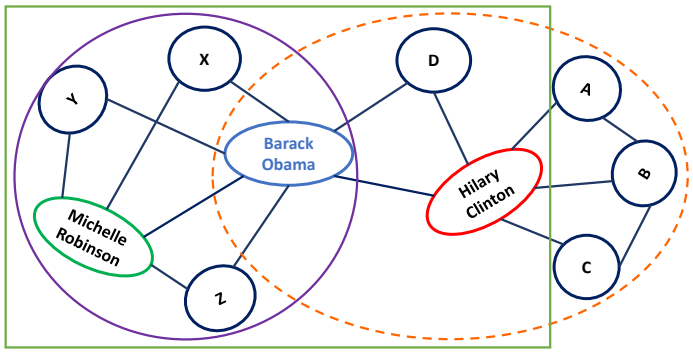

Fig. 1. Community graph

Fig. 1 shows an example of such type of graph where the entity of a query, its type and relationship name are *Barack Obama*, *person* and *spouse* accordingly. The candidate responses are *Michelle Robinson* and *Hilary Clinton*. The objective is to classify *Michelle Robinson* as the correct response based on their community analysis. The communities of *Barack Obama* (green rectangle), *Michelle Robinson* (purple circle) and *Hilary Clinton* (orange ellipse) are defined by *IN_SAME_SENTENCE* relation which means the pair of entities are mentioned in the same sentence in texts. The graph is thus constructed from untyped semantic relationships based on co-occurrences. It would also be possible to use typed semantic relationships provided by a relation extraction system or a knowledge base.

3.2 Construction of the Community Graph

The graph of entities as illustrated in Fig. 1 is created from a graph representing the knowledge extracted from the texts (lower part of Fig. 2) called knowledge graph. The knowledge graph represents documents, sentences, mentions and entities as nodes and the edges between these nodes represent relationships between these elements. This knowledge graph is generated after applying systems of named entity recognition (NER) and sentence splitting.

Recognition of named entities is done by using Stanford system [26] and *Luxid* of ExpertSystem[4]. Luxid is a rule-based NER system that uses some external information sources such as Freebase, geo-names etc and performs with high precision. It decomposes the entity mentions into components, such as *first name*, *last name* and *title* for a *person* named entity and classifies *location* named entities into *country*, *state/province* and *city*. When the two NER systems disagree, as in (Stanford: location, Luxid: person), we choose the annotation produced by Luxid because it provides more precise information about the detected entity than Stanford does.

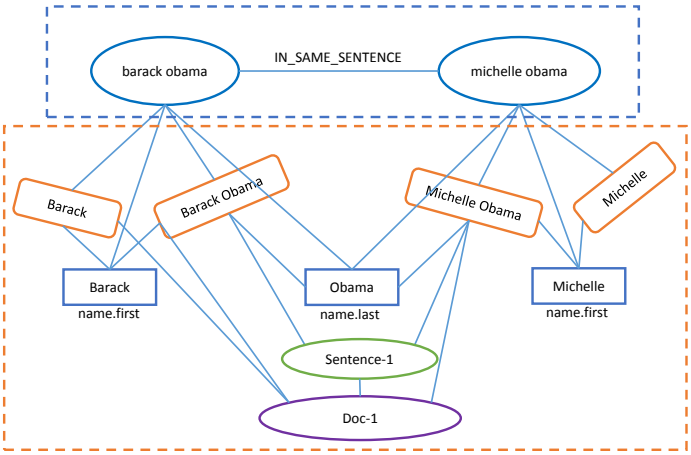

Fig. 2. Knowledge graph

Multiple mentions of the same entity found in the same document are connected to the same entity node in the knowledge graph, based on the textual similarity of the references and their possible components, which corresponds to a first step of entity linking on local criteria. This operation is performed by Luxid. However, an entity can be mentioned in different documents also with different forms (e.g. *Barack Obama*, *President Barack Obama*, *President Obama* etc.) which creates redundant nodes in the knowledge graph. Entities are then grouped according to the similarity of their names and the similarity of their neighboring entities calculated by Eq. 4. This step groups the

[4] `http://www.expertsystem.com/`

similar entities into a single node in the community graph (upper part of Fig. 2). This latter graph is constructed from the information on the entities and relations present in the knowledge graph and the link with the documents is always maintained. It is thus possible to know the number of occurrences of each entity and each relation. The graph is stored in a Neo4j database, a graph-oriented database, which makes it possible to extract the subgraphs linked to an entity by queries. We only consider as members of the communities the entities of type person, location and organization.

4 Relation Validation

In order to predict whether a relationship is correct or not, we consider this problem as a binary classification task based on three categories of information. We calculate a set of features using the graphs (see section 4.1), to which we add features based on a linguistic analysis of the text that justifies the candidate and describes the relationship (see section 4.2) and an estimation of trust on the candidates (voting) according to the frequencies of them in the responses of each query.

4.1 Graph-based Features

We explore information at entity level based on community graph analysis. We assume that a true object is an important member in the community of the subject entity. A community X_e of a subject is defined by the sub-graph formed by its neighbors up to several levels. A merging of the communities of two particular entities includes all the neighbors of that pair of entities. We, therefore, define different features related to this hypothesis. We compute 4 features on the community graph a) network density b) eigenvector centrality c) mutual information and d) network similarity.

Network density (Eq. 1) computes the degree of connectivity among the nodes of the network. A network gets high density score if there are many connections among the neighbor nodes.

$$\rho_{X_e} = \frac{number\,of\,existing\,edges\,with\,e}{number\,of\,possible\,edges} \tag{1}$$

We hypothesize that the density of the community of a true object merged with the community of the subject entity must be higher than the density of a false object community merged with that of the same subject because the subject and true object shares more neighbors that makes more edges among the network nodes. According to the Fig. 1 the merged community of *Michelle Robinson* and *Barack Obama* is more dense than the merged community of *Hilary Clinton* and *Barack Obama*.

Eigenvector centrality [27] measures the influence of a node in a graph. A node will be even more influential if it is connected to other influential nodes. We hypothesize that the subject would be more influenced by a true object than by a false object since the true object shares more community members with the subject and becomes highly influential. We measure the influence of an object in the community of the subject by calculating the absolute difference between the eigenvector centrality scores of the subject and object. We assume that this difference should be smaller for a true object than

for a false object because the subject and the true object should get similar score according to their influence to each other. Suppose $A = (a_{i,j})$ is the adjacency matrix of a graph G. The eigenvector centrality x_i of node i is calculated recursively by Eq. 2.

$$x_i = \frac{1}{\lambda} \sum_k a_{k,i} x_k \tag{2}$$

where, $\lambda \neq 0$ *is a constant and the equation can be expressed in matrix form:* $\lambda x = xA$

Mutual information quantifies the amount of information gained by a random variable compare to another random variable. We compute mutual information gained by the community of an object through the community of the subject to capture the evidence of having relationship between them by using the Eq. 3.

$$MI(X_q, X_c) = H(X_q) + H(X_c) - H(X_q, X_c) \tag{3}$$

$$\textit{where, } H(X) = -\sum_{i=1}^{n} p(e_i)\ log_2(p(e_i))$$

$$and\, p(e) = \frac{number\ of\ edges\ of\ e}{number\ of\ edges\ of X}$$

$and\ X_q\ and\ X_c\ are\ the\ communities\ of\ the\ subject\ and\ object\ entity\ respectively,$
$and\ p(e)\ is\ the\ probability\ of\ degree\ of\ centrality\ of\ a\ community - member.$

We hypothesize that the mutual information of a true object should be higher than a false object because its community shares more edges to the community of the subject.

Network similarity computes the similarity between two communities in term of common neighbors by using Eq. 4.

$$similarity = \frac{|X_q \cap X_c|}{\sqrt{|X_q||X_c|}} \tag{4}$$

$where,\ X_q\ and\ X_c\ are\ the\ community\ members\ of\ the\ subject\ and$
$object\ entity\ accordingly.$

The similarity gets higher score if both communities share large number of neighbors. Thus we hypothesize that the similarity of the communities of a subject and a true object should be higher than with a false object.

However, sometimes the value of the network density or similarity between the communities of a subject and a false object may be higher than that between the subject and a true object. For example, the network density or similarity score between *Barack Obama* and *Hillary Clinton* can be higher than the score between *Barack Obama* and *Michelle Obama* based on the existence of other members of their networks in the corpus. Moreover, the same pair of entities may have multiple relations that cannot be distinguished by graph analysis. Basically, graph based features are useful to compute the degree of association between the pairs of entities but these features do not hold the semantics of different relation types. Therefore, we need to define some linguistic features for characterizing the true meaning of relations.

4.2 Linguistic Features

We analyze two kinds of linguistic features for RE: syntactic and semantic. Syntactic features are able to give some clues for assessing if a relation exists between a pair of mentions. The semantics of a relation is captured by trigger words and thus we will analyze the sentence at lexical level.

Syntactic dependency analysis facilitates to compute the syntactic features, i.e, the parser [26] provides a tree in which nodes are the words of the sentence and the edges between them are labeled by their syntactic role. The consecutive dependency labels between a pair of mentions in a sentence form a pattern of the relation that is expressed by the sentence. Such pattern can be repeated for expressing the same relationship between a different pair of mentions.

We extract a list of dependency patterns for each relation and simplify them as we did in [8]. However, it is hard to capture all the dependency patterns since the relations are expressed in many different ways. Therefore, we compute minimal edit distance of an unknown pattern to the known patterns so that the unknown pattern can be considered as a member of one of the known pattern groups by an approximation. We propose the minimum edit distance as a feature and call it dependency pattern edit distance (DPED). Since relations are often expressed in short dependency paths, the length of the simplified dependency path is considered as a feature.

The semantic analysis is performed based on positive triggers associated to the relation types. Positive triggers refer to the keywords that strongly characterize a particular relation. For example, *wife, husband, married* are positive triggers for a *spouse* relation.

Since the relations are expressed by a variety of words, it is hard to collect all the positive triggers for a relation. Therefore, we associate a word embedding to each trigger by using the *GloVe*[5] model. Thus, deciding if a word is a trigger or not relies on the similarity of their embeddings. Suppose, *[a, b]* are two words between the subject and object mentions in a sentence and *[x, y, z]* the pre-collected positive triggers for the claimed relation. We compute the cosine similarity between the vectors of each pair of *[(a,x), (a,y), (a,z), (b,x), (b,y), (b,z)]* and take the best similarity score as a feature. If a word between the pair of mentions completely matches to one of the pre-collected positive triggers the similarity score is 1.0. We inspect the existence of any positive trigger in three cases: 1) between the mentions at surface level 2) in the dependency path and 3) in the minimum subtree as in [15]. We define the baseline feature set by combining the path length of simplified dependency with these three features.

5 Data

This section describes two kinds of data. Firstly, data that have been used for computing the linguistic and graph features and secondly, the datasets used for training and testing several models.

[5] https://nlp.stanford.edu/projects/glove/

5.1 Data for Computing Features

For linguistic features, we use the 2014 assessed corpus of TAC-KBP English cold start slot filling (CSSF) which contains examples of correct relations for around 38 relationships between $1,020$ entity pairs. Trigger words and dependency patterns (as discussed in section 4.2) of different relations were collected from this dataset. We collected the words between the subject and object pairs of positive responses of each kind of relation and ranked them by counting their frequencies. We observed that for some relations (i.e. *spouse*) the top 5 words are discriminating (i.e. *married*) for characterizing the semantics of the relation. We call these relations trigger-dependent relations. In contrast, we notice that for other relations (i.e. *city_of_residence*) top 5 words are either prepositions or other words that are not able to distinguish any semantic relation. Such relations are called trigger-independent relations. Thus we selected 12 trigger-dependent relations (first column of Table 1) for our study. We obtain in total 76 trigger words and 286 patterns for these relations.

For computing the entity graph as discussed in section 3.2, we used the reference corpus of TAC-KBP CSSF evaluation. We parsed around $50,000$ and $30,000$ documents provided for the CSSF-2015 and CSSF-2016 evaluation tasks accordingly. Both corpus included texts from newswire and discussion forums. Two knowledge graphs have been built from these datasets and there are $152,583$ and $65,389$ entities (*person, organization* and *location*) in the 2015 and 2016 graphs accordingly. Moreover, the knowledge graphs consist of $805,216$ and $488,198$ edges of *IN_SAME_SENTENCE* relations among different entity mentions accordingly.

5.2 Data for Training and Testing the Models

In our experiments, we use subsets of TAC-KBP CSSF datasets of 2015 and 2016 for training and testing accordingly. The CSSF task requires a participant system to respond to a set of queries. Each query is about an entity (subject), associated with the slot (relationship) to fill. A system responses to a query by providing an object value, an object type, an object offset, the relation provenance offset and a confidence score. The relation provenance is an excerpt of a document that justifies the claimed relation. The relation provenance offsets of a response is not guaranteed to delimit a complete sentence. Thus we extract the complete sentence corresponding to the relation provenance offset snippet from the source document as several features have to be computed on the complete sentences.

A lot of queries have been answered with only wrong responses by different systems. Therefore, we keep queries that have been answered with at least one correct response. We also filter out queries that are not relevant to the relations we study in our experiments. In order to build a set of positive and negative examples for training and testing, we extracted answers corresponding to those queries from the systems assessment files. An assessment file contains the indication whether an object and a relation provenance text of a query relation are correct or not. There are many wrong responses to the queries regarding the amount of correct ones. Therefore, we reduce the number negative examples to construct a balanced training dataset. We randomly select a subset of wrong responses from each query of the training dataset. After removing the

duplicate responses, the ratio of positive and negative responses is around $2/3$. Similar process of extracting positive and negative examples has been applied for building the test dataset but we do not filter any negative example.

Since graph based features depends on the performance of NER systems that could be not good enough to detect all the named entities mentioned in the queries and responses, our system could compute graph features for a small number of responses. In order to compute features on the graphs, we defined two strategies i.e. *hard constraint* and *relaxed constraint* in terms of connectivity in a graph between the subject and object entities of a relation under inspection.

Hard Constraint: Usually, a relation between two entities is expressed when both of the entities are mentioned in the same sentence. Therefore, in our preliminary study, we constrained the system to find an IN_SAME_SENTENCE link (in the knowledge graph) between the subject and object entities of a relationship under observation for computing graph features. Thus we could compute graph features for around 14% of the responses. We obtained $2,274$ (827 positive and $1,447$ negative) instances for training from 130 queries of the 12 trigger-dependent relations. In this setting, the test dataset counted $3,429$ (262 positive and $1,167$ negative) instances from 63 queries for the same number of relations. The number of training instances (positive and negative) for different relations are very different. Moreover, some relations (i.e. *per:spouse, org:member_of*) count very small number of training instances and some (i.e. *per:children, country_of_death*) have no training example at all. Therefore, in the setting of hard constraint we include training examples of some other relations to the instances of the 12 selected relations. We obtained in total $3,481$ ($1,268$ positive and $2,213$ negative) instances from 260 queries of 19 relations. Our experiment on this dataset obtained poor result (it will be discussed in section: 6.1) because of small number of training examples. Therefore, we defined a strategy for increasing the number of examples.

Relaxed Constraint: We relax the constraint of IN_SAME_SENTENCE between the subject and object entities of a relation. If the entity pairs are not connected by an IN_SAME_SENTENCE link in the graph we forcefully connect them by creating the link before computing graph features and delete the link after completing the feature computation of the entity pair. Thus our system could compute graph features for around 50% of the responses. Relaxed constraint significantly increases both training and test instances for all the relations as shown in Table 1. We obtain a training dataset that counts in total $14,804$ ($5,933$ positive and $8,871$ negative) instances from 411 queries of the 12 trigger-dependent relations. In similar way, our test dataset counts $1,109$ and $4,827$ positive and negative instances accordingly from 223 queries.

6 Results and Discussion

We performed experiments on the 12 trigger-dependent relations as discussed in section 5.1. We select these relations to measure the effect of the proposed graph-based features when they are combined with linguistic features.

	Hard Constraint						Relaxed Constraint					
Relation Name	**# Train Data**		**# Test Data**		**F**	**Acc.**	**# Train Data**		**# Test Data**		**F**	**Acc.**
	Pos.	**Neg.**	**Pos.**	**Neg.**			**Pos.**	**Neg.**	**Pos.**	**Neg.**		
per:parents	35	12	17	15	66.7	65.6	148	229	94	386	50.0	77.1
per:children	0	0	0	221	0.0	99.1	67	93	37	630	65.9	95.7
per:spouse	2	1	0	0	-	-	155	298	25	106	49.1	79.4
per:country_of_death	0	0	6	114	53.3	94.2	77	148	72	189	88.3	93.1
per:country_of_birth	14	28	1	65	8.0	65.2	108	140	5	260	80.0	99.3
per:city_of_death	56	100	0	0	-	-	243	398	30	227	51.4	86.0
per:city_of_birth	141	281	70	22	86.8	77.2	485	814	139	90	91.8	90.4
per:employee_or_member_of	211	355	16	232	15.4	73.4	2,517	3,267	287	1,538	50.4	80.8
org:top_members_employees	68	78	29	30	96.7	96.6	461	743	61	277	63.4	79.9
org:member_of	8	14	0	0	-	-	571	917	27	389	58.1	91.4
org:country_of_headquarters	158	310	82	334	23.0	74.3	471	822	140	362	54.3	78.9
org:city_of_headquarters	134	268	41	134	44.6	58.9	630	1,002	192	373	74.7	83.0
All Together	827	1,447	262	1,167	51.5	78.2	**5,933**	**8,871**	**1,109**	**4,827**	**63.3**	**84.8**

Table 1. Comparison of relation validation performance between hard and relaxed constraints

We evaluate our features for characterizing a relation in the setting of a relation validation (RV) task. The RV method takes as input a text snippet, subject, object and the claimed relation name and outputs true if the snippet holds the claimed relation otherwise false. We include a voting feature (as done in [8]) to the linguistic and graph features to observe the trustworthy influence of multiple systems on validating a claimed relation.

Furthermore, we evaluate the contribution of our RV model for a knowledge base population (KBP) task that will be discussed in Section 6.2. Both tasks RV and KBP are evaluated by standard precision, recall and F-score.

6.1 Results on the Relation Validation Task

In this section, we want to observe the effectiveness of adding more training data on relation validation task, performance of different classifiers for binary classification and impact of the proposed features on relation extraction which is evaluated as a relation validation task.

We want to inspect the efficiency of relaxed constraint of IN_SAME_SENTENCE link between subject and object entities over hard constraint to improve the classification performance. We expect that the relation validation system would learn and perform better by training with and testing on more data accordingly.

Table 1 represents the statistics of training and test dataset, F-score (F) and accuracy (Acc.) regarding both hard and relaxed constraints. This table shows the scores obtained by Random Forest classifier which is trained by the best feature combination, i.e. $Voting+Linguistic+Graph$. Relaxed constraint significantly increases both training and test instances for all the relations as discussed in section 5.2. The F-score and accuracy are gained over almost all the relations by relaxing the IN_SAME_SENTENCE constraint. In the results on hard constrained dataset, we notice that several relations do

Classifier	Precision	Recall	F-measure	Accuracy
LibLinear	45.6	73.9	56.1	79.1
SVM	49.8	73.5	59.4	81.6
Naive Bayes	48.0	76.5	59.0	80.6
MaxEnt	48.3	69.5	57.0	80.6
Random Forest	**57.8**	**70.1**	**63.3**	**84.8**

Table 2. Relation validation performances by different classifiers

not have any training examples (as *per:children* or *per:country_of_death*), or test data (as *per:spouse, per:city_of_death, org:member_of*). We see that relaxed constraint results better F-score for all the relations except *per:parents* and *org:top_members_employees*. We obtain overall F-score and accuracy of 51.5 and 78.2 accordingly by hard constraint. In contrast, the relaxed constraint improves these scores by around 12 and 6 points accordingly. We achieve overall F-score and accuracy of 63.3 and 84.8 accordingly by relaxing the constraint. Thus relaxation of IN_SAME_SENTENCE constraint between subject and object entities facilitates to train a model with more data that significantly improves the performance to classify the correct and wrong relations.

We also compare the classification performances of different classifiers e.g. LibLinear, SVM, Naive Bayes, MaxEnt and Random Forest based on the best feature combination (Voting+Linguistic+Graph) on the dataset of relaxed constraint as shown in Table 2. We achieve the best precision (57.8), F-score (63.3) and accuracy (84.8) by Random Forest classifier although it gets a lower recall (70.1) compared to other classifiers. The best recall of 76.5 is resulted by Naive Bayes which obtains the third highest F-score (59.0) and accuracy (80.6).

However, it may not be system-independent to use the voting feature to realize a task of relation extraction. Usually, a relation extractor does not employ multiple systems for generating relation hypothesis. Therefore, we discard the voting feature in this RV evaluation to realize the contribution of proposed features for relation extraction. We define a baseline (BL) by four linguistic (semantic and syntactic) features as discussed in Section 4.2 and observe relation validation performances through the linguistic baseline, the proposed linguistic and graph features and their combinations. Since Random Forest results the best score over several classifiers, we observe the performances of different feature sets by this classifier.

Table 3 represents the classification scores where we observe that the combination of BL and proposed graph features outperforms the BL almost for all the relations except *per:country_of_death*. We obtain overall F-score of 58.60 by BL+Graph that is around 9 points higher than the BL. The experimental results also show that the combination of BL and dependency pattern edit distance (DPED) improves the overall F-score by 1.79 point over the BL. This combination achieves higher F-score for 7 relations (among 12) which indicates the effectiveness of DPED for RV task. Basically, we gain higher precision by allowing a slight drop of recall that results better F-score over the BL. The best F-score is achieved by the combination of BL, DPED and graph

	BL			BL + DPED			BL + Graph			BL + DPED + Graph		
Relation Name	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
per:parents	37.30	73.40	49.46	40.12	69.15	50.78	51.75	78.72	**62.45**	45.65	67.02	54.31
per:children	62.22	75.68	68.29	60.87	75.68	67.47	70.00	75.68	72.73	93.33	75.68	**83.58**
per:spouse	36.23	100	53.19	73.53	100	84.75	65.00	52.00	57.78	68.42	52.00	**59.09**
per:country_of_death	98.55	94.44	**96.45**	98.55	94.44	**96.45**	97.50	54.17	69.64	98.55	94.44	**96.45**
per:country_of_birth	11.43	80.00	20.00	12.90	80.00	22.22	100	80.00	**88.89**	100	80.00	**88.89**
per:city_of_death	58.00	96.67	72.50	71.05	90.00	79.41	75.00	90.00	**81.82**	75.00	90.00	**81.82**
per:city_of_birth	97.20	100	98.58	97.18	99.28	98.22	100	99.28	**99.64**	100	99.28	**99.64**
per:employee_or_member_of	20.74	29.27	24.28	20.63	27.53	23.58	34.85	24.04	28.45	32.88	25.09	**28.46**
org:top_members_employees	39.39	63.93	48.75	35.78	63.93	45.88	52.38	90.16	66.27	59.14	90.16	**71.43**
org:member_of	28.57	44.44	34.78	38.71	44.44	41.38	48.00	44.44	46.15	50.00	44.44	**47.06**
org:country_of_headquarters	52.17	25.71	34.45	59.02	25.71	35.82	75.93	29.29	**42.27**	75.93	29.29	**42.27**
org:city_of_headquarters	50.00	44.27	46.96	60.14	43.23	50.30	86.67	47.4	**61.28**	89.69	45.31	60.21
All Together	44.75	**55.73**	49.64	48.55	54.46	51.34	65.09	53.29	58.60	**66.02**	54.82	**59.90**

Table 3. Classification performances by different feature sets

(BL+DPED+Graph). This combination results overall F-score of 59.90 which is around 10 points higher than the BL. We observe that BL+DPED+Graph obtains higher F-score for 11 relations compare to the BL. For only one relation (*per:country_of_death*) the classification performance remains same as the BL.

We notice in Table 3 that BL+Graph and BL+DPED+Graph obtain a surprising performance for *per:country_of_birth* over the BL. Both BL+Graph and BL+DPED+Graph achieve an F-score of 88.89 which is around 69 points higher than the BL. The reason behind this result is that we have a very small number of true instances for this relation compare to the number of false instances (as shown in Table 4) and a high precision is resulted by discarding large number of false relations.

We achieve the highest precision almost for all the relations by BL+DPED+Graph. BL+DPED+Graph achieves overall precision of 66.02 that is around 21 points higher than the BL that indicates the proposed features discard large number of false relation instances correctly. A little drop of recall is caused by BL+DPED+Graph which is around 1 point less than the BL. The recall of 55.73 and 54.82 are resulted by the BL and BL+DPED+Graph accordingly. The drop of recall indicates the limitations of graph features to hold the semantic evidences of some relations.

Table 4 illustrates the confusion matrix resulted by BL and BL+DPED+Graph where we compare the number of true positive (TP), false negative (FN), false positive (FP), true negative (TN) and accuracy (Acc.). We see that the baseline and BL+DPED+Graph methods correctly classify overall 618 and 608 true relation instances accordingly among $1,109$. That means BL+DPED+Graph discards 501 true relation instances which is around 1% more than the BL. However, the BL and BL+DPED+Graph correctly discard overall $4,064$ and $4,514$ false relation instances respectively among $4,827$. The rate of discarding false relation instances by BL+DPED+Graph is around 9% higher than the BL which contributes to increase the overall precision and finally achieves a high accuracy. While observing the accuracy relation-by-relation we see a significant improvement achieved by BL+DPED+Graph over the BL for all the relations.

	BL					BL + DPED + Graph				
Relation Name	**TP**	**FN**	**FP**	**TN**	**Acc.**	**TP**	**FN**	**FP**	**TN**	**Acc.**
per:spouse	25	0	44	62	66.41	13	12	6	100	**86.28**
per:parents	69	25	116	270	70.62	63	31	75	311	**77.92**
per:children	28	9	17	613	96.10	28	9	2	628	**98.35**
per:country_of_death	68	4	1	188	98.08	68	4	1	188	**98.08**
per:country_of_birth	4	1	31	229	87.92	4	1	0	260	**99.62**
per:city_of_death	29	1	21	206	91.44	27	3	9	218	**95.33**
per:city_of_birth	139	0	4	86	98.25	138	1	0	90	**99.56**
org:top_members_employees	39	22	60	217	75.74	55	6	38	239	**86.98**
org:member_of	12	15	30	359	89.18	12	15	12	377	**93.51**
org:country_of_headquarters	36	104	33	329	72.71	41	99	13	349	**77.69**
org:city_of_headquarters	85	107	85	288	66.02	87	105	10	363	**79.65**
per:employee_or_member_of	84	203	321	1217	71.29	72	215	147	1391	**80.10**
All Together	618	491	763	4064	78.87	608	501	313	4514	**86.29**

Table 4. Comparison of the confusion matrices resulted by BL and BL+DPED+Graph

Claimed Relation	**Justification Sentence**	**RV**
spouse(Willem-Alexander, Maxima Zorreguieta Cerruti)	**Willem-Alexander** married **Maxima Zorreguieta Cerruti** from Argentina and they have three daughters: Princess Catharina-Amalia, Princess Alexia and Princess Ariane.	TP
children(Margaret Thatcher, Mark)	In a statement to the public, **Thatcher**'s son **Mark** Thatcher said his twin sister Carol and the rest of their family had been overwhelmed by messages of support they had received from around the globe.	TP
spouse(Willem-Alexander, Alexia)	**Willem-Alexander** married Maxima Zorreguieta Cerruti from Argentina and they have three daughters: Princess Catharina-Amalia, Princess **Alexia** and Princess Ariane.	TN
children(Margaret Thatcher, Carol)	In a statement to the public, **Thatcher**'s son Mark Thatcher said his twin sister **Carol** and the rest of their family had been overwhelmed by messages of support they had received from around the globe.	FN

Table 5. True positive (TP), true negative (TN) and false negative (FN) examples after validating relations

Table 5 presents classification results on some claimed relations from the test data that helps to realize the performance of our RV model. The first and second row show two correctly classified true claims of *spouse* and *children* relation accordingly. Furthermore, a false claim of *spouse* relation has been detected as wrong as shown in the third row. In contrast, our system fails to correctly classify a true *children* relation as shown in the fourth row. However, our system achieves overall descent scores compared to the baseline. All the experimental results on RV task show that global information about the entities captured by the community-graph based features are significantly effective for RE task.

6.2 Results of Knowledge Base Population Task

One objective of RE is the population of knowledge base. Since existing RE systems generate a large number of wrong relationships, it is interesting to know whether the validation step allows for building a better KB.

	Precision	**Recall**	**F-score**
System-1	36.73	22.78	28.12
System-2	32.07	24.89	28.03
System-3	37.50	21.52	27.35
Voting+Linguistic+Graph	**38.51**	**24.05**	**29.61**
Linguistic+Graph	29.53	18.57	22.80
Voting	24.88	21.10	22.83

Table 6. KBP performances by some top ranked systems (upper part) and our RV models (lower part)

For evaluating KBP task, we define a *ground truth* (GT) for all the queries that contains different correct objects for each of the queries. An object is considered as correct if the excerpt containing the subject and object justifies their relation, otherwise wrong. A system should not repeat an answer (object) for the same query. If a system repeats an object for the same query only one instance of that object would be considered as correct and others would be wrong. Moreover, there are some queries such as *city_of_birth* whose object should be a single value. Therefore, a system has to response with a single object for such query. In our KBP system, we select an object randomly if several candidates are validated as correct for such relation. We compute the KBP performances of the single systems that participated to the TAC KBP evaluation on our test dataset for comparison. The test dataset given by the TAC KBP organizers provides the assessments of the slot filling responses of all the participating systems. Therefore it allows us to compute their results on the subset of queries of our test set. The top 3 TAC KBP systems on our test set individually obtained F-score of 28.12, 28.03 and 27.35 accordingly (see upper part of Table 6).

Since different relation extraction systems can be employed for the KBP task, we can use the *voting* feature to take advantage of the agreements on the outcomes by several relation extraction systems. Therefore, we built a RV model by using a single voting feature. Since the best performance of RV is achieved by Voting+Linguistic+Graph features, we use the RV model trained by this feature combination for the KBP task.

In the lower part of Table 6, we see that the voting based KBP system obtains an F-score of 22.83 which indicates the importance of this feature. Interestingly, our Voting+Linguistic+Graph based KBP system achieves a F-score of 29.61 which is higher than each individual KBP system. We also observe that Voting+Linguistic+Graph based KBP system achieves the highest precision of 38.51 that is almost 2 points higher than the best KBP system. The precision improvement indicates that our model discards many wrong relations which are resulted by different RE systems. Moreover, Voting+Linguistic+Graph based KBP system obtains the recall of 24.05 that is around 1.27 point higher than the best RE based KBP system and around 3 points higher than the voting based KBP system. These results justify that our system enables to fill more relations in knowledge base than the existing ones specially for the trigger dependent relations.

7 Conclusion

In this paper, we have presented community-based features for RE task that are able to capture some global information about the entities in a relationship. The proposed features are computed on a community graph extracted from a corpus and they measure how two entities are associated globally when they are in a relationship. Since such kind of measurements cannot characterize the semantics of relations, we combine these with some linguistic features that are able to characterize the type of a relation. We have shown that the proposed graph based features significantly improve the performance of relation extraction over the baseline. The proposed features also enable to globally select a large number of true values for populating a knowledge base and helps to obtain better scores over the sate-of-the-art system for the relations we studied.

One of our objectives is now to explore graph algorithm to exploit our graph representation for relation validation and KBP tasks in an unsupervised fashion.

References

1. GuoDong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: Proceedings of the 43rd annual meeting on association for computational linguistics, Association for Computational Linguistics (2005) 427–434
2. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2005) 724–731
3. Fundel, K., Küffner, R., Zimmer, R.: Relex—relation extraction using dependency parse trees. Bioinformatics **23**(3) (2007) 365–371
4. Nguyen, T.H., Grishman, R.: Relation extraction: Perspective from convolutional neural networks. In: VS@ HLT-NAACL. (2015) 39–48
5. Augenstein, I.: Web Relation Extraction with Distant Supervision. PhD thesis, University of Sheffield (2016)
6. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: a graph-based method. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM (2011) 765–774
7. Friedl, D.M.B., Heidemann, J., et al.: A critical review of centrality measures in social networks. Business & Information Systems Engineering **2**(6) (2010) 371–385
8. Rahman, R., Grau, B., Rosset, S.: Community graph and linguistic analysis to validate relationships for knowledge base population. In: 4th Annual International Symposium on Information Management and Big Data (SIMBig). (2017) 133-143
9. Zhou, G., Zhang, M., Ji, D., Zhu, Q.: Tree kernel-based relation extraction with context-sensitive structured parse tree information. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). (2007)
10. Jiang, J., Zhai, C.: A systematic exploration of the feature space for relation extraction. In: HLT-NAACL. (2007) 113–120
11. Gamallo, P., Garcia, M., Fernández-Lanza, S.: Dependency-based open information extraction. In: Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP, Association for Computational Linguistics (2012) 10–18

12. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics (2011) 541–550
13. Mooney, R.J., Bunescu, R.C.: Subsequence kernels for relation extraction. In: Advances in neural information processing systems. (2006) 171–178
14. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, Association for Computational Linguistics (2009) 1003–1011
15. Chowdhury, M.F.M., Lavelli, A.: Combining tree structures, flat features and patterns for biomedical relation extraction. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2012) 420–429
16. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Machine Learning and Knowledge Discovery in Databases. Springer (2010) 148–163
17. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: 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, Association for Computational Linguistics (2012) 455–465
18. Vu, N.T., Adel, H., Gupta, P., Schütze, H.: Combining recurrent and convolutional neural networks for relation classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Association for Computational Linguistics (June 2016) 534–539
19. Zheng, S., Xu, J., Zhou, P., Bao, H., Qi, Z., Xu, B.: A neural network framework for relation extraction: Learning entity semantic and relation pattern. Knowledge-Based Systems **114** (2016) 12–23
20. Dligach, D., Miller, T., Lin, C., Bethard, S., Savova, G.: Neural temporal relation extraction. EACL 2017 (2017) 746
21. Gardner, M., Mitchell, T.M.: Efficient and expressive knowledge base completion using subgraph feature extraction. In: EMNLP. (2015) 1488–1498
22. Wang, Q., Liu, J., Luo, Y., Wang, B., Lin, C.: Knowledge base completion via coupled path ranking. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. (2016) 1308–1318
23. Yu, D., Ji, H.: Unsupervised person slot filling based on graph mining. In: ACL. (2016)
24. Guo, Y., Che, W., Liu, T., Li, S.: A graph-based method for entity linking. In: IJCNLP, Citeseer (2011) 1010–1018
25. Holzinger, A., Ofner, B., Stocker, C., Valdez, A.C., Schaar, A.K., Ziefle, M., Dehmer, M.: On graph entropy measures for knowledge discovery from publication network data. In: Availability, reliability, and security in information systems and HCI. Springer (2013) 354–362
26. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. (2014) 55–60
27. Bonacich, P., Lloyd, P.: Eigenvector-like measures of centrality for asymmetric relations. Social networks **23**(3) (2001) 191–201