Community graph and linguistic analysis to validate relationships for knowledge base population

Rashedur Rahman, Brigitte Grau, Sophie Rosset

Community graph and linguistic analysis to validate relationships for knowledge base population

Rashedur Rahman[1] and **Brigitte Grau**[2] and **Sophie Rosset**[3]
[1] IRT SystemX, LIMSI, CNRS, Université Paris-Saclay
rashedur.rahman@irt-systemx.fr
[2] LIMSI, CNRS, ENSIIE, Université Paris-Saclay
brigitte.grau@limsi.fr
[3] LIMSI, CNRS, Université Paris-Saclay
sophie.rosset@limsi.fr

Abstract

Relation extraction between entities from text plays an important role in information extraction and knowledge discovery related tasks. Relation extraction systems produce a large number of candidates where many of them are not correct. A relation validation method justifies a claimed relation based on the information provided by a system. In this paper, we propose some features by analyzing the community graphs of entities to account for some sort of world knowledge. The proposed features improve validation of relations significantly when they are combined with voting and some state-of-the-art linguistic features.

1 Introduction

Extracting relations from texts is important for different information extraction and knowledge discovery related tasks such as knowledge base population, question-answering, etc. This task requires Natural Language Understanding of pieces of text which is particularly complex when searching for a large number of semantic relations that describe entities in the open domain. The relationship types can be relative to the family of a person (*spouse, children, parents* etc.) or characteristics of a company (*founder, top_members_or_employees* etc.), etc. This task, named slot filling, is evaluated in the KBP evaluation[1] in which systems must extract instances of around 40 relation types related to several kinds of entities (person, organization, location and their different sub-types). Thus, in order to take advantage of several system's capabilities and improve results, a final step can be added that enables to validate results of the systems. The method described in this paper is in the framework of the latter task which, given an entity and a response provided by a system (its value and a text-segment that justifies the claimed relation), has to decide whether the value is correct or not. We focus on relations that occur between two entities.

Different approaches have been studied for validating relations particularly by evaluating the confidence that a system can have on the source of the response, i.e. the document that justifies the response (Yu et al., 2014) and the confidence score of the system (Viswanathan et al., 2015). Nevertheless, other criteria are needed that concern validating semantics of a relation by linguistic characteristics (Niu et al., 2012; Hoffmann et al., 2011; Yao et al., 2011; Riedel et al., 2010) and are similar to those used in relation extraction task.

However, in most cases, the different relation validation methods do not take into account the global information, that can be computed on the collection of text-documents. Collection level global information about the object of a relation and words around the mentions have been taken into account for web relation extraction by (Augenstein, 2016). Such information allows to introduce some sort of world knowledge for making choices based on criteria that are independent of how a relation is expressed in the text-segment. We hypothesize that two entities having a true relationship should be linked to more common entities than a proposed false relationship between that pair of entities. For example, the spouse of a person will share more places and relationships with his/her spouse than with other people. Therefore, we extracted a graph of entities from the collection that allowed us to propose new characterizations of the relations by graph-based features (Han et al., 2011), (Friedl et al., 2010), (Solá et al., 2013). We also introduce information-theoretic measurements on the graph of entities, some of

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

which have been successfully used in other tasks, such as entropy for knowledge detection in publishing networks (Holzinger et al., 2013) and mutual information for the validation of responses in question answering systems (Magnini et al., 2002; Cui et al., 2005). Additionally, we propose dependency pattern edit-distance for capturing the syntactic evidence of relations. Word-embeddings have also been explored to detect the unknown triggers of relation expression.

The relation validation method we propose is thus based on three categories of information: linguistic information associated with the expression of the relations in texts, information coming from the graphs of entities built on the collection, and finally information related to the systems and the proposals made. We evaluated our relation validation system on a sub-part of KBP CSSF-2016 corpus and show that the validation step achieves around 5% to 8% higher accuracy over the baseline features when they are combined together with the proposed graph-based features.

2 Related Works

Relation validation methods have studied different kinds of features to decide if a type of relation exists or not.

Existence and semantic assessment of relation candidates rely on linguistic features, as syntactic paths or the existence of trigger words between the pair of entity-mentions. Dependency tree (Culotta and Sorensen, 2004), (Bunescu and Mooney, 2005), (Fundel et al., 2007) provides clues for deciding the presence of a relation in unsupervised relation extraction. Gamallo et al. (2012) proposed rule-based dependency parsing for open information extraction. They defined some patterns of relation by parsing the dependencies and discovering verb-clauses in the sentences.

Syntactic analysis cannot characterize the type of a relation. Therefore, words around the entity mentions in sentences have been analyzed to characterize the semantics of a relation (Niu et al., 2012), (Hoffmann et al., 2011), (Yao et al., 2011), (Riedel et al., 2010), (Mintz et al., 2009). Chowdhury et al. (2012) proposed a hybrid kernel by combining dependency patterns and trigger words for bio-medical relation extraction. Thus we explored these different kinds of linguistic features for validating relationships.

It can be difficult to identify the trigger words for different types of relation in the open domain. Therefore, recently neural network based methods have been popular for relation classification task (Vu et al., 2016), (Dligach et al., 2017), (Zheng et al., 2016). These methods use word-embeddings for automatically learning the patterns and semantics of relations without using any handcrafted features. Dependency based neural networks have also been proposed (Cai et al., 2016), (Liu et al., 2015) to capture features on the shortest path.

A voting method has been proposed by (Sammons et al., 2014) for ensemble systems to validate the outcomes that are proposed by multiple systems from different information sources. This method shows good results and remains stable from a dataset to another.

Several graph based methods (Gardner and Mitchell, 2015), (Lao et al., 2015), (Wang et al., 2016) have been proposed for knowledge base completion task by applying Path Ranking Algorithm (Lao and Cohen, 2010). These methods basically use the already existing relationships in a knowledge base to learn inference and create new relations by the inference model. Yu and Ji (2016) proposed a graph based method for trigger-word identification for slot filling task by using PageRank and Affinity propagation on a graph built at sentence level.

Information-theoretic measurements on graphs have been successfully used in some related tasks. Holzinger et al. (Holzinger et al., 2013) measured entropy to discover knowledge in publication networks. Some question-answering systems measured point-wise mutual information (Magnini et al., 2002), (Cui et al., 2005) to exploit redundancy. In order to find the important and influential nodes in a social network, centralities of the nodes have been measured (Friedl et al., 2010). Solá et al. (2013) explored the concept of eigenvector centrality in the multiplex networks. In order to validate the proposed relationships, we apply these different measures on graphs of entities constructed from the text-collection.

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 re-

lation extraction systems. Suppose, other relations $r_o \epsilon R$ where $r_o \neq r_q$. We characterize whether a candidate-entity $e_c i$ of E_c is correct or not for a query relation (r_q) by analyzing the communities of 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.

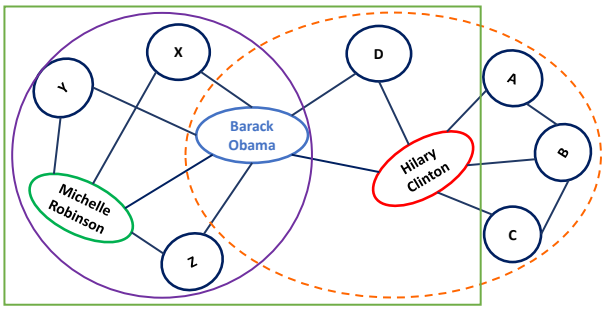

Figure 1: Community graph

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* that are linked to *Barack Obama* by *spouse* relation hypothesis. The objective is to classify *Michelle Robinson* as the correct response based on the 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 the text. 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.

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. This knowledge graph is generated after applying systems of named entity recognition (NER) and sentence splitting.

Recognition of named entities is done using Stanford system (Manning et al., 2014) and *Luxid*[2]. Luxid is a rule-based NER system that uses some external information sources such as Freebase, geo-names etc and perform with high precision. It is able to decompose 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 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.

The knowledge graph represents documents, sentences, mentions and entities as nodes and the edges between these nodes represent relationships between these elements.

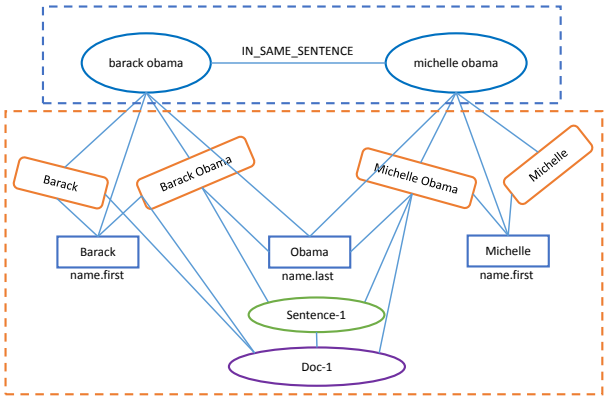

Figure 2: Knowledge graph

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 with different forms (eg, *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. 2. This step groups the similar entities into a single entity 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

[2] http://www.expertsystem.com/fr/

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 according to the frequencies of them in the responses of each query (see section 4.3). Table 1 summarizes all the features used for the classification task.

4.1 Graph-based Features

We assume that a correct candidate of a query is an important member of the community of the query entity. A community X_e of an entity 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 hypothesize that the *network density* (Eq. 1) of the community of a correct candidate merged with the community of the query entity must be higher than the density of an incorrect candidate's community merged with that of the same entity.

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

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*.

We compute the *network similarity* (Eq. 2) between two communities and hypothesize that the score of the network similarity between the communities of a query entity and a correct candidate would be higher than the score between that query entity and a wrong candidate.

$$similarity = \frac{|X_q \cap X_c|}{\sqrt{|X_q||X_c|}} \quad (2)$$

where, X_q and X_c are the community members of the query entity and of a candidate entity accordingly.

The *eigenvector centrality* (Bonacich and Lloyd, 2001) measures the influence of a node in a network. A node will be even more influential if it is connected to other influential nodes. We hypothesize that the query-entity should be more influenced by the correct candidate than by other candidates. We measure the influences of the candidates in the community of the query-entity by calculating the absolute difference between the score of eigenvector centrality of the query-entity and that of each candidate. We, therefore, assume that this difference should be smaller for a correct candidate than for an incorrect candidate. 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. 3.

$$x_i = \frac{1}{\lambda} \sum_k a_{k,i} x_k \quad (3)$$

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

We also hypothesize that *mutual information* (Eq. 4) between the pair of communities of a query-entity and a correct candidate must be higher than that computed between the communities of the query-entity and an incorrect candidate.

$$MI(X_q, X_c) = H(X_q) + H(X_c) - H(X_q, X_c) \quad (4)$$

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

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

X_q and X_c are the communities of a query-entity and a candidate respectively.

The community of an entity (query-entity or candidate) is extended up to the third level to measure eigenvector centrality and mutual information.

4.2 Linguistic Features

For assessing if a relation exists between the pair of entity mentions, we define syntactic features. For characterizing the semantic of the relation, we represent it by seed words and analyze the sentence at the lexical level.

Syntactic features are calculated from dependency analysis, i.e. the parser (Manning et al., 2014) provides a tree in which nodes are the

Feature Group	Feature Name
Graph	Network density
	Eigenvector centrality
	Mutual information
	network similarity
Linguistic	Minimum edit distance between dependency patterns
	Dependency pattern length
	Are the query and filler mentions found in the same clause
	Has trigger word between mentions
	Has trigger word in dependency path
	Has trigger word in minimum subtree
	Is trigger based relation
Baseline (voting)	Filler credibility

Table 1: Relation validation features

words of the sentence and the edges between them are labeled by their syntactic role. We collected a list of dependency patterns for each relation from annotated examples. For example, in the sentence *Paola, Queen of the Belgians is the wife of King Albert of Belgium.* the dependency pattern between *Paola* and *King Albert* is *[nn, nsubj, prep_of]* and the dependencies are *nn(Queen, Paola), nsubj(wife, Queen), prep_of(wife, Albert)*. We simplify the pattern *[nn, nsubj, prep_of]* to *[nsubj, prep_of]* by removing leading and following *nn* for noise reduction. We notice that sometimes the dependency patterns contain consecutive labels like *[nsubj, dobj, prep_of, prep_of, poss]*. In such cases, we simplify the pattern by substituting the consecutive labels with a single label which leads to simplify *[nsubj, dobj, prep_of, prep_of, poss]* into *[nsubj, dobj, prep_of, poss]*. This simplification generalizes the dependency patterns.

The acquired patterns are compared to the simplified dependency path of a sentence by computing edit distances. Suppose a list of pre-annotated dependency patterns are (a,b,c), (a,c,d), (b,c,d) for a relation R and the dependency pattern (a,c,b) is extracted from a relation provenance sentence between the query and the filler mention for a claimed relation to be R. We calculate the edit distance between each pair of [(a,c,b), (a,b,c)], [(a,c,b), (a,c,d)],[(a,c,b), (b,c,d)] and keep the minimum edit distance as a feature.

Since relations are often expressed in short dependency paths, the length of the simplified pattern is considered as a feature.

The semantic analysis is performed based on trigger words associated with the relation types. We consider semantic features as boolean values by defining two types of trigger words: positive trigger and negative trigger. Positive trigger words refer to the keywords that strongly support a particular relation while the negative triggers strongly negate the claimed relations. For example, *wife, husband, married* are positive triggers while *parent, children, brother* are negative triggers for a *spouse* relation. We collected these seed words from the assessed dataset of TAC KBP 2014 slot filling task. In total we collected around 250 triggers and 553 dependency patterns of 41 relations from $3,579$ annotated snippets.

Since the relations are expressed by a variety of words it is hard to collect all the trigger words for a relation. Therefore, we associate a word embedding to each trigger by using a pre-trained *GloVe* (Pennington et al., 2014) 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 query and filler mention of a claimed relation *R* and *x, y, z* the positive triggers for the claimed relation. We compute the similarity between the vectors of each pair of *(a,x), (a,y), (a,z), (b,x), (b,y), (b,z)*. If the similarity score for

a word from *a, b* satisfies a predefined threshold (0.7) we consider that there exists a trigger word. We check whether there is any positive and/or negative trigger word in three cases for validating a claimed relation: 1) between the mentions at surface level 2) in the dependency path and 3) in the minimum subtree as in (Chowdhury and Lavelli, 2012).

Some relations can be expressed without using any trigger word. For example, the snippet *Mr. David, from California won the prize* expresses the *per:city_of_residence* relation without explicitly using any trigger word. We classify the relation types in two classes: can be expressed without trigger word or not, and use a boolean flag (*is_trigger-based_relation*) as a feature.

4.3 Voting: Filler Credibility

We use and calculate the credibility score for candidates based on all the responses given by different systems to a query.

$$filler\ credibility\ (F_i, Q) = \frac{number\ of\ occurrences\ of\ F_i}{\#\ of\ occurrences\ of\ all\ the\ candidates} \quad (5)$$

Let F be the candidates of a query Q supplied by the systems S. The credibility of a candidate F_i is computed by the equation 5.

The filler credibility counts the relative vote of a candidate which indicates the degree of agreement by different systems to consider the candidate as correct. Since we can assume that systems already performed some linguistic and probabilistic analysis to make the responses, filler credibility holds strong evidence for a candidate to be correct. Some slot filling and slot filler validation methods have used the system credibility (Yu et al., 2014) and confidence score (Wang et al., 2013; Viswanathan et al., 2015; Rodriguez et al., 2015) of the responses for validating relations but these features are much system and data dependent. Therefore, we use only filler credibility as the baseline.

5 Experiments and Results

5.1 Data

We perform our experiments by using TAC-KBP English cold start slot filling (CSSF) datasets of 2015 and 2016. TAC provided a reference corpus for the English CSSF-2015 evaluation task that consisted of $45,000$ documents. These documents include texts from newswires and discussion_forums. We parsed these texts for building the knowledge graph. We compiled our training data from the assessments of English CSSF-2015 responses of slot filling systems. There were $9,339$ round-1 queries for English CSSF2015 and in total $330,314$ round-2 queries were generated by all the slot filling systems based on the responses of round-1 queries. NIST assessed SF responses of around $2,000$ round-1 and $2,500$ round-2 queries. A lot of queries have been answered with only wrong responses. Therefore, we do not take into account these queries for building our training corpus. We selected only queries that have been answered with correct and wrong responses. This subset counted total $1,296$ ($1,080$ round-1 and 216 round-2) slot filling queries.

We extracted answers corresponding to those queries from the system assessment file that contains the assessment of the filler values and relation-provenance offsets accordingly. The relation provenance offsets refer to the document ids and begin-end position of the text segments in the evaluation corpus. The values of filler assessment can be correct (C), wrong (W) and inexact (X) while the assessment values of relation provenance can be correct (C), wrong (W), short (S) and long (L) where S and L are considered as inexact. We only take into account the C and W filler assessments and separate the correct and wrong responses according to the relation provenance assessment. When the relation provenance assessment is C the filler assessment can be either C or X but not W. It results in $68,076$ responses. Several features have to be computed on complete sentences, and not on sentence excerpts. As the relation provenance offset of a SF response is not guaranteed to be a complete sentence, we extract the complete sentence corresponding to the relation provenance offset snippet from the source document.

The linguistic features are calculated from the analyzed sentences where the mentions of the pair of entities (query and candidate) must be identified. However, our system cannot find both entries in all selected sentences. This happens when either the query entity or the candidate entity is mentioned by a pronoun or nominal anaphora as

we do not use any co-reference resolution. In addition, the named entity detection system, which results from two efficient systems, does not detect all the entity mentions (of queries and candidates) present in the queries and hypotheses. This restriction also applies to the computation of features based on the graph that is constructed over the recognized named entities. This behavior corresponds to a trend generally observed in named entity recognition systems when applied to different documents from those on which they were trained (here web documents and blogs instead of newspaper articles or Wikipedia pages).

Moreover, adding the constraint to finding two entities in the same sentence causes this additional decrease in performance. A total of $55,276$ hypotheses (of the initial $68,076$) could be processed to compute the linguistic features for the responses of $1,296$ queries that have been responded with both correct and wrong candidates. Our system restricts to extract graph based features of limited number of query-responses due to the NER limitation and in_same_sentence constraint. In summary, we can extract both the linguistic and graph features for $4,321$ responses from 260 queries, (213 from round-1 and 47 from round-2). Since there are many wrong responses compared to the number of correct responses, we take a subset of the wrong responses randomly from CSSF-2015 dataset for training the system after removing the duplicate responses where the ratio of correct and wrong responses of each query is $2:1$ approximately. After applying the filtering process the training dataset contains in total $3,481$ ($1,268$ positive and $2,213$ negative) instances.

Similar process has been applied on TAC KBP CSSF-2016 dataset that we use for testing. CSSF-2016 dataset consists of around $30,000$ documents. Around $34,267$ responses (of 925 queries) have been assessed as correct or wrong by TAC. Our system could compute graph based features for around $3,884$ responses of 352 queries that have been responded by both correct and wrong answers. There are 699 correct and $3,185$ wrong responses among $3,884$ responses with graph based features in our test dataset. The statistics of the training and test datasets are shown in the columns 1 to 4 of Table 4.

5.2 Results

We have trained the models by using several classifiers and evaluated relation validation method by standard precision, recall, F-measure and accuracy of different models.

In this experiment we show the contribution of the proposed graph based features for validating relations. Since community-graph based analysis does not account the semantics but holds some evidences of how the entities are associated to each other we expect significant gain of precision by our relation validation method so that a better F-score can be achieved.

We compare the classification performances of different classifiers e.g. LibLinear, SVM, Naive Bayes, MaxEnt and Random Forest based on the best combination of the features as shown in Table 2. We obtain the best precision (32.2), recall (48.1), F-score (38.5) and accuracy (72.4) by Random Forest classifier. The second highest precision (29.0) and accuracy (72.35) are resulted by MaxEnt while Naive Bayes results the second highest recall (45.8) and F-score (38.5). Since Random Forest results the best scores over other classifiers we observe the performances of different feature sets by this classifier.

Table 3 presents the classification performances of different feature sets by Random Forest classifier. Filler credibility as a single feature obtains the F-score and accuracy of 34.6 and 62.1 accordingly. It represents a strong baseline, as shown in (Sammons et al., 2014). The combination of filler credibility and linguistic features boosts the performance by around 6 points in term of accuracy though the F-score drops slightly. When the filler credibility is combined with the graph features it gains a significant precision (29.0) and accuracy (70.3) which are around 4 and 8 points higher accordingly. This combination also increases the F-score slightly. This may seem surprising, as the graphs are the same for assumptions about pairs of identical entities but linked by different relationships. It should be noted that the relation hypotheses are already the results of different systems based on the semantics of relations. The graph-based features account the global context of the hypotheses of relations and the experimental results show the significant contribution of them.

The best precision (32.2), F-score (38.5) and accuracy (72.4) are achieved by combining filler credibility, linguistic and graph features. This

Classifier	Precision	Recall	F-measure	Accuracy
LibLinear	26.5	38.5	31.4	69.72
SVM	23.7	29.9	26.4	70.06
Naive Bayes	28.0	45.8	34.8	69.07
MaxEnt	29.0	37.1	32.5	72.35
Random Forest	**32.2**	**48.1**	**38.5**	**72.40**

Table 2: Relation validation performances by different classifiers

Feature Groups	Precision	Recall	F-measure	Accuracy
Filler credibility (Fc)	25.1	**55.8**	34.6	62.1
Fc + Linguistic	26.8	45.2	33.7	68.0
Fc + Graph	29.0	45.1	35.3	70.3
Fc + Linguistic + Graph	**32.2**	48.1	**38.5**	**72.4**

Table 3: Relation validation performances by different feature sets

combination improves the precision, F-score and accuracy by around 7, 4 and 10 points over the filler credibility. The precision is gained because of low false negative that indicates the system classifies a small number of wrong responses as correct. This results strongly signify the contribution of graph based features for validating the claimed relations.

We also investigated the classification performance of *Fc+Linguistic+Graph* model relation by relation as shown in Table 4.

We notice that the classification performances are not similar for different relations according to the F-score and accuracy although we train a single model with the responses of different relations. However, as we do not have the similar number of training instances for all the relations (e.g. *per-city_of_birth* and *org-subsidiaries*), as we see in column 2, it may impact the results. Additionally, in the test data, there are a very small number of positive responses compared to the negative ones (see column 5) for some relations that cause inconsistency in classification performance. For example, the distribution of positive and negative responses of *per:top_members_employees*, *per:city_of_birth* and *per:parents* are more balanced compared to *countries_of_residence*, *org-subsidiaries*, *employee_or_member_of*, *country_of_headquarters*. Therefore, these two sets of relations make a clear difference between their scores. Also, some relations (*org-parents*, *per-country_of_birth* and *per-stateorprovince_of_birth*) have very small number of positive instances where all of these have been classified as wrong and these relations individually counts zero true positive. Therefore, the precision, recall and F-score become zero. However, most of the negative instances of these relations have been correctly classified as wrong. Moreover, the test dataset contains only negative instances for some relations (*per-cities_of_residence*, *per-schools_attended*, *per-children*, *org-alternate_names* and *org-founded_by*). All of the instances have been correctly classified as wrong that result 100% accuracy for these relations. Interestingly, we notice that proposed relation validation method discards all the wrong instances of *per-children* relation although the training dataset does not contain any positive or negative instances of this relation. A similar result has also been observed for *per-country_of_death* relation where the system of relation validation obtains an accuracy of 86.7 to validate the instances. These results justify that the relation validation system trained by the instances of different relations is able to predict correctly whether an instance of a relation is correct or wrong even though the system is not trained by the instances of that particular relation.

Table 5 explains better that our system performs better than the baseline system to discard the negative responses. It presents the confusion matrix

	Training Data		Test Data			
Relation Name	**# Instances**	**Pos. (%)**	**# Instances**	**Pos. (%)**	**F-score**	**Accuracy**
org-top_members_employees	147	46.3	59	29 (49.2%)	93.1	93.2
per-city_of_birth	423	33.3	92	70 (76.1%)	86.8	77.2
statesorprovinces_of_residence	81	33.3	170	70 (41.2%)	78.8	80.0
org-city_of_headquarters	402	33.3	175	41 (23.4%)	63.0	80.6
per-parents	37	94.6	32	17 (53.1%)	61.1	56.3
per-country_of_death	0	0	120	6 (5%)	33.3	86.7
per-countries_of_residence	501	33.3	1461	261 (17.9%)	27.9	66.0
org-country_of_headquarters	474	33.3	416	82 (19.7%)	22.0	71.0
stateorprovince_of_headquarters	157	31.2	369	60 (16.3%)	11.4	74.8
org-subsidiaries	60	35	251	34 (13.5%)	15.4	65.0
per-employee_or_member_of	566	37.3	248	16 (6.5%)	11.9	64.1
org-parents	10	40	21	10 (47.6%)	0.0	52.4
per-country_of_birth	42	33.3	66	6 (1.5%)	0.0	65.2
per-stateorprovince_of_birth	93	33.3	19	2 (10.5%)	0.0	52.6
per-cities_of_residence	78	33.3	123	0 (0%)	0.0	85.4
per-schools_attended	12	33.3	34	0 (0%)	0.0	100
per-children	0	0	221	0 (0%)	0.0	100
org-alternate_names	65	49.2	6	0 (0%)	0.0	100
org-founded_by	40	70	1	0 (0%)	0.0	100
All Together	3481	36.4	3884	699 (18%)	38.5	72.4

Table 4: Performance of relation validation relation by relation

Feature Groups	**TP**	**FN**	**FP**	**TN**
Filler credibility (Fc)	390 (55.8%)	309	1,164	2,021 (63.5%)
Fc + Linguistic	316 (45.2%)	383	862	2,323 (72.9%)
Fc + Graph	315 (45.1%)	384	771	2,414 (75.8%)
Fc + Linguistic + Graph	336 (48.1%)	363	709	2,476 (77.7%)

Table 5: Confusion matrix resulted by different feature sets (where the number of positive and negative instances are 699 and $3,185$ accordingly)

(true positive (TP), false negative (FN), false positive (FP) and true negative (TN)) of the classification task by different feature sets. The combination, *Fc+Linguistic+Graph* discards 77.7% wrong responses which is around 14% higher than the baseline. All the experimental results signify the contribution of the proposed features for the task of relation validation.

6 Conclusion

This paper presents a method of validating relationships from the system outputs. We have introduced some features on the linked entities which are computed at the global level of the collection. We have proposed to deal with the community graphs of entities that make it possible to account for general knowledge about the entities having true relationships. Experimental results have shown that our proposed features significantly improve a baseline constructed from the votes on the responses of different systems. The proposed method outperforms the baseline to discard wrong relationships.

The calculation of the different characteristics is dependent on the parsing of the texts, in particular, on the results of the NER system. This part has to be improved in order to evaluate the contribution of community graph on more responses. Although the proposed method results better F-score and accuracy compared to the baseline, the method also discards some positive responses that drops the recall; thus we have to overcome this limitation.

References

Isabelle Augenstein. 2016. *Web Relation Extraction with Distant Supervision*. Ph.D. thesis, University of Sheffield.

Phillip Bonacich and Paulette Lloyd. 2001. Eigenvector-like measures of centrality for asymmetric relations. *Social networks* 23(3):191–201.

Razvan C Bunescu and Raymond J Mooney. 2005. 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, pages 724–731.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 756–765. http://www.aclweb.org/anthology/P16-1072.

Md Faisal Mahbub Chowdhury and Alberto Lavelli. 2012. 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, pages 420–429.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 400–407.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 423.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. *EACL 2017* page 746.

Dipl-Math Bettina Friedl, Julia Heidemann, et al. 2010. A critical review of centrality measures in social networks. *Business & Information Systems Engineering* 2(6):371–385.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex—relation extraction using dependency parse trees. *Bioinformatics* 23(3):365–371.

Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. 2012. Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*. Association for Computational Linguistics, pages 10–18.

Matt Gardner and Tom M Mitchell. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. In *EMNLP*. pages 1488–1498.

Xianpei Han, Le Sun, and Jun Zhao. 2011. 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, pages 765–774.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. 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, pages 541–550.

Andreas Holzinger, Bernhard Ofner, Christof Stocker, André Calero Valdez, Anne Kathrin Schaar, Martina Ziefle, and Matthias Dehmer. 2013. On graph entropy measures for knowledge discovery from publication network data. In *Availability, reliability, and security in information systems and HCI*, Springer, pages 354–362.

Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning* 81(1):53–67.

Ni Lao, Einat Minkov, and William W Cohen. 2015. Learning relational features with backward random walks. In *ACL (1)*. pages 666–675.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng WANG. 2015. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 285–290. http://www.aclweb.org/anthology/P15-2047.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. Is it the right answer?: exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 425–432.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. 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, pages 1003–1011.

Feng Niu, Ce Zhang, Christopher Ré, and Jude W Shavlik. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS* 12:25–28.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

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

Miguel Rodriguez, Sean Goldberg, and Daisy Zhe Wang. 2015. University of florida dsr lab system for kbp slot filler validation 2015. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

Mark Sammons, Yangqiu Song, Ruichen Wang, Gourab Kundu, Chen-Tse Tsai, Shyam Upadhyay, Siddarth Ancha, Stephen Mayhew, and Dan Roth. 2014. Overview of ui-ccg systems for event argument extraction, entity discovery and linking, and slot filler validation. *Urbana* 51:61801.

Luis Solá, Miguel Romance, Regino Criado, Julio Flores, Alejandro Garcia del Amo, and Stefano Boccaletti. 2013. Eigenvector centrality of nodes in multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 23(3):033131.

Vidhoon Viswanathan, Nazneen Fatema Rajani, Yinon Bentor, and Raymond Mooney. 2015. Stacked ensembles of information extractors for knowledge-base population. In *Proceedings of ACL*.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. 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*. Association for Computational Linguistics, San Diego, California, pages 534–539. http://www.aclweb.org/anthology/N16-1065.

I-Jeng Wang, Edwina Liu, Cash Costello, and Christine Piatko. 2013. Jhuapl tac-kbp2013 slot filler validation system. In *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*. volume 24.

Quan Wang, Jing Liu, Yuanfei Luo, Bin Wang, and C Lin. 2016. Knowledge base completion via coupled path ranking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. pages 1308–1318.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1456–1466.

Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, Shi Zhi, Jiawei Han, Clare R Voss, and Malik Magdon-Ismail. 2014. The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In *COLING*. pages 1567–1578.

Dian Yu and Heng Ji. 2016. Unsupervised person slot filling based on graph mining. In *ACL*.

Suncong Zheng, Jiaming Xu, Peng Zhou, Hongyun Bao, Zhenyu Qi, and Bo Xu. 2016. A neural network framework for relation extraction: Learning entity semantic and relation pattern. *Knowledge-Based Systems* 114:12–23.