Matrix Models with Feature Enrichment for Relation Extraction

Duc-Thuan Vo(✉) and Ebrahim Bagheri

Laboratory for Systems, Software and Semantics (LS3),
Ryerson University, Toronto, Canada
{thuanvd, bagheri}@ryerson.ca

Abstract. Many traditional relation extraction techniques require a large number of pre-defined schemas in order to extract relations from textual documents. In this paper, to avoid the need for pre-defined schemas, we employ the notion of *universal schemas* that is formed as a collection of patterns derived from Open Information Extraction as well as from relation schemas of pre-existing datasets. We then employ matrix factorization and collaborative filtering on such universal schemas for relation extraction. While previous systems have trained relations only for entities, we exploit advanced features from relation characteristics such as clause types and semantic topics for predicting new relation instances. This helps our proposed work to naturally predict any tuple of entities and relations regardless of whether they were seen at training time with direct or indirect access in their provenance. In our experiments, we show improved performance compared to the state-of-the-art.

Keywords: Matrix factorization · Universal schema · Relation extraction · Topic modeling

1 Introduction

Relation Extraction (RE) aims at determining the relationships between entities in textual documents and is among the more important tasks of information extraction that has been applied in a large number of applications such as question-answering, and search engines, among others. In this context, most supervised and semi-supervised extraction methods use a predefined, finite and fixed schema of relation types (such as located-in or founded-by). Among the supervised methods, the works in [8, 21] have focused on performing language analysis for semantic relation extraction. A running theme among these techniques is the capacity to generate linguistic features based on syntactic, dependency, or shallow semantic structures of the text. Semi-supervised approaches have been employed by various researchers [7, 12, 18] to extract patterns derived initially from rule-based relations. These approaches exploit the concept of *information redundancy* and hypothesize that similar relations tend to appear in uniform contexts. However, most of these systems are limited in terms of scalability and portability across domains by predefined and fixed schema of relation types.

In contrast, Open Information Extraction (OIE) [5, 6, 11, 19] systems offer a more nuanced approach that rely minimally on background knowledge and manually labeled

M. Mouhoub and P. Langlais (Eds.): Canadian AI 2017, LNAI 10233, pp. 235–247, 2017.
DOI: 10.1007/978-3-319-57351-9_28

training data. OIE systems require no supervision for performing highly scalable extractions and are often portable across domains. Distant supervision [1, 15, 16] aims to exploit information from knowledge bases such as Freebase in order to learn large-scale relations from text. Heuristic method [16] has been employed to generate training relations by mapping phrases to their corresponding entities in KBs. Dependence on pre-existing datasets in distant supervision approaches can be avoided by using language itself as the source for the universal schema. To this end, Riedel et al. [15] have already presented a model based on matrix factorization with universal schemas for predicting relations. These authors presented a series of models that learn lower dimensional manifolds for tuple of entities and relations with a set of weights in order to capture direct correlations between relations. While these approaches have shown reasonable performance, their limitation is in that they train cells only for tuple of entities, and therefore, are limited when an insufficient number of evidences are present for the entities present in the relations. For instance, the relation OBAMA–president-of–US could not infer the hidden relation HOLLANDE-president-of-FRANCE due to differences of the tuples <(OBAMA, US)> and <(HOLLANDE, FRANCE)>. Even if entity types are exploited in the system, the failure to predict other relations such as OBAMA–born-in–US with similar tuple of entities can be problematic in such systems.

In this paper, we exploit advanced features from relation characteristics, namely *clause types* and *semantic topics* to enrich the cells in the matrix of a matrix factorization model for predicting new relation instances. Particularly, we exploit clause types and topic models to predict relations regardless of whether they were seen at training time with direct or indirect access. Our work uses the concept of universal schema from [15] in order to convert the KB combined with OIE patterns into a binary matrix in which tuples of entities are its rows and relations denote the columns.

The rest of this paper is organized as follows. Section 2 presents background on relation extraction with matrix factorization and collaborative filtering. In Sect. 3, we present a detailed description of several models with feature enrichments in matrix models. This is followed by an in-depth discussion of experimental results in Sect. 4, where the results are compared to the state-of-the-art. Section 5 finalizes the paper with conclusions and future work.

2 Background

The application of matrix factorization and collaborative filtering methods in relation extraction aims at predicting hidden relations that might not have been directly observed. Kemp et al. [9] used Infinite Relational Model (IRM) in order to build a framework to discover latent relations jointly from an n-dimensional matrix. In this matrix, each dimension has a latent structure through which relations can be found. Bollegala et al. [3] try to explore clusters of entity pairs and patterns jointly as latent relations by employing co-clustering. Takamatsu et al. [17] use probabilistic matrix factorization with Singular Value Decomposition to reduce dimensions to discover relations. Riedel et al. [15] use matrix factorization and collaborative filtering by including surface patterns in a universal schema and a ranking objective function to

learn latent vectors for tuple of entities and relations. In their systems, they use surface patterns extracted from OIE. The goal of these systems is to predict the hidden relations through *matrix completion*. Our work is similar to [15] in that we use matrix factorization and collaborative filtering for the discovery of potential relations. Given the fact that the work in [15] populate the matrix cells only for entity pairs, they can fall short when predicting latent relations that do not have sufficient evidence from observed relation instances. In our work, we represent universal schemas in the form of a matrix where tuples of entities form the rows and relations constitute the columns. We further employ advanced features from relation characteristics such as clause types and semantic topics for enriching the cells in the matrix to predict new relation instances (See Fig. 1).

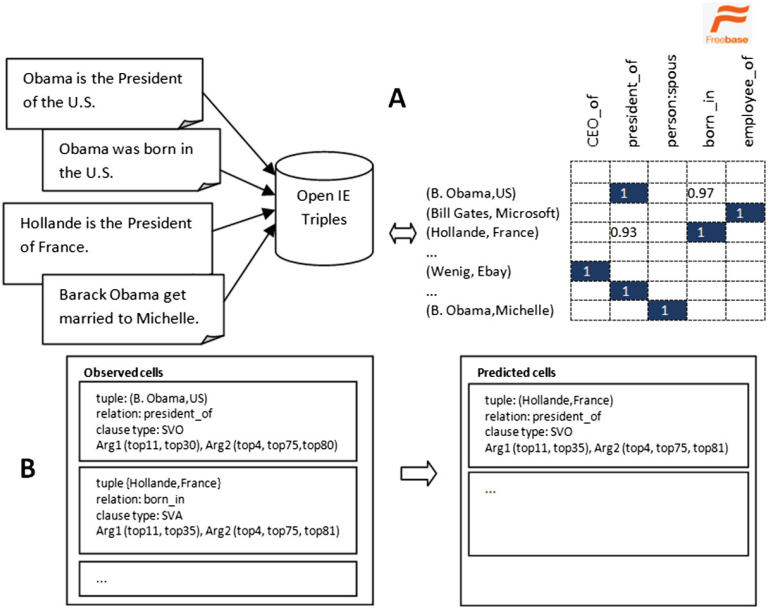

Fig. 1. (A) Universal schema with relation and tuples, 1 denotes observed relation, values 0.97, 0.93 are predicted probability of the relation; (B) Examples of observed cells and predicted cells.

Open Information Extraction (OIE) [5, 6, 11, 19] is another closely related area of research to our work. The majority of OIE systems use a shallow syntactic representation or dependency parsing in the form of verbs or verbal phrases and their arguments. Wu and Weld [20] propose a shallow syntactic representation of natural language text in the form of verbs or verbal phrases and their arguments. Besides that, there have been several approaches [11, 18, 20] that employ robust and efficient dependency parsing for relation extraction. More recent OIE systems such as ClausIE [5] use dependency parsing and a small set of domain-independent lexica without any post-processing or training data. At the outset, these systems exploit linguistic knowledge about the grammar of the English language to first detect clauses in an input

sentence and to subsequently identify each clause type based on the grammatical function of its constituents. In our work, for surface patterns of the matrix, we use OIE patterns for populating relations of all kinds in the matrix. We exploit clause-based features extracted from OIE (ClausIE) combined with topic models (LDA), which are used as important characteristics for predicting potential relations.

The work in distant supervision [1, 15, 16] aim at exploiting knowledge bases (KBs) such as Freebase to learn relations. Heuristic method [16] is employed to generate training relations by mapping pairs of mentioned entities in a sentence to corresponding entities in a KB. As a result, such methods do not require labeled corpora, avoid being domain dependent, and allow the use of any size of documents. These methods learn extracted relations for a known set of relations. Universal schema [15] employs the notion of distant supervision by using a knowledge base to derive similarity between both structured relations such as "LocatedAt" and surface form relations such as "is located at" extracted from text. Factorization of the matrix with universal schemas results in low-dimensional factors that can effectively predict unseen relations. Our work is close to [15] in that we convert the KB into a binary matrix with tuple of entities corresponding to the rows and relations corresponding to the columns in the matrix.

3 The Proposed Approach

Riedel et al. [15] have presented a universal schema to build a matrix, which is a union of patterns extracted by OIE from text and fixed relations from knowledge bases. In our work, we use clause-based OIE [5] to extract surface patterns with fully-enriched clause feature structures. Our task is to predict the hidden relations by completing the schema in the matrix over surface patterns and fixed relations. Using the same notation as [15], we use T and R to correspond to entity tuples and relations. Given a relation $r \in R$ and a tuple $t \in T$, the objective of our work is to derivea fact or relation instance about a relation r and a tuple of two entities $e1$ and $e2$. A matrix is constructed with size $|T| \times |R|$ for relation instances. Each matrix cell presents a fact as $x_{r,t}$ and is a binary variable. The variable in each cell of the matrix is 1 when relation r is true for the tuple t, and 0 when relation r is false for t. We aim at predicting new relations that could potentially hold for tuple of entities, which are missing in the matrix. We present several models that can address the task as follows.

3.1 Matrix Factorization (F Model)

In the matrix factorization approach, we denote each relation by a_r and each tuple of entities as e_t. We measure compatibility between relation r and tuple t as the dot product of two latent feature representations of size k. Thus we have:

$$\theta_{r,t} = \sum_k a_{r,k} e_{t,k} \tag{1}$$

The formula is factorizing a matrix into a multiplication of two matrices $\Theta = AE$, A denoting the lower dimension matrix of a_r, and E representing the lower dimension matrix of e_t based on PCA [4]. Thus, a model with the matrix $\Theta = (\theta_{r,t})$ of natural parameters is defined as the low rank factorization AE. To estimate the values in PCA, we have:

$$x_{r,t} = \sigma(\theta_{r,t}) = \sigma\left(\sum_k a_{r,k} e_{t,k}\right) \tag{2}$$

This is applying a logistic function $\sigma(\theta_{r,t}) = 1/(1 + \exp(-\theta_{r,t}))$ [4, 14, 15] to model a binary cell in the matrix. Each cell is drawn from a Bernoulli distribution with natural parameter $\theta_{r,t}$. We maximize the log-likelihood of the observed cells under a probabilistic model to learn low dimensional representations. The representations a_r and e_t can be found by minimizing the negative log-likelihood using stochastic gradient descent with $x_{r,t} = \sigma(\theta_{r,t})$. This formulation also applies to all the following models as well.

3.2 Neighbor Model (N Model)

In the matrix, a relation in a column could be neighbor to some other co-occurring relation (neighbor relation). For example, relation "CEO-of" and "Director-of" are often seen in similar relation instances. Therefore, the Neighbor Model [10] is essential to capture the localized correlation of the cells in the matrix to incorporate this information. We implement a neighborhood model N via a set of weights w of features based on co-occurrence of information around tuples of entities, e.g., headword "President" often appears in tuples of entities in relations such as "CEO-of" and "Director-of". In this model, each cell is scored based on the set of weights between this cell and its associated neighbors. This leads to the following formulation:

$$\theta_{r,t} = \sum_k w_k f_k(r', r) \tag{3}$$

where w_k is the weight of association between r' and r; $f_k(r', r)$ defines a conjunctive feature between relation r and neighbor relation r' and k is the number of relations r' that have the exact same tuples as r.

In this model, we additionally employ clause-based features, which are core characteristic of relations for selectional preference. For instance, a relation OBAMA-president-of-US or OBAMA-leader-of-US could be presented by a clause type "Subject-Verb-Complement", while another relation "OBAMA-born-in-US" is in the form of a "Subject-Verb-Adverb" clause. Therefore, considering only entities will fail to predict relations in tuple <(OBAMA, US)>. We have used clause types in OIE [5] when extracting surface patterns for the matrix. We can interpolate the confidence for a given tuple and relation based on the trueness of other similar relations for the same tuple. Measuring compatibility of an entity tuple and relation amounts to summing up the compatibilities between each argument slot representation and the corresponding entity

representation. We extend the model in Eq. 3 to incorporate clause types, which is presented as follows:

$$\theta_{r,t} = \sum_{k} v_{t,r} w_k f_k(r', r) \tag{4}$$

where $v_{t,r}$ is a vector of clause types.

3.3 Entity Model (E Model)

Earlier, Riedel et al. [15] introduced the use of entities in collaborative filtering. In their method, they employed entities to predict latent relations. The model embeds each entity into a low dimensional space of size k. For binary relations, their arguments (t_1 and t_2) are entities modeled in the low dimensional space and are represented as $e1$ for t_1 and $e2$ for t_2. The equation below leads to the calculation of the compatibility of tuple of entities and their relations by summing up the presentation of each argument slot. Thus, this leads to:

$$\theta_{r,t} = \sum_{k} a_{t1,k} e1_{r,k} + \sum_{k} a_{t2,k} e2_{r,k} \tag{5}$$

Analogous to the Neighbor model, we augment the entity model with clause-based features, which enhances the entity model as follows:

$$\theta_{r,t} = \sum_{k} a_{t1,k} e1_{r,k} v_{t1} + \sum_{k} a_{t2,k} e2_{r,k} v_{t2} \tag{6}$$

where v_{t1} isa vector of clause type for argument 1, and v_{t2} is a vector of clause type for argument 2.

3.4 Topic-Based Model (T Model)

In the Entity model, selectional preferences are employed based on each argument's slot representation and the corresponding entity representation in order to learn from other relations. However, in addition to this, many relations can be considered to be related to other relations based on the probability of being observed within the same semantic topic group. For instance, the relation tuple <(HOLLANDE, FRANCE)> could be learned from the observed relation <OBAMA, US>, if and when "OBAMA"-"HOLLANDE" and "US"-"FRANCE" are observed in the same semantic topic groups. Therefore, relations could further be learned by their selectional preferences in semantic topic groups. This helps to determine more relations that are missing when learning from directly observed relations. We use Latent Dirichlet Allocation [2] to generate semantic groups of topics, and then embed this information in the matrix. We embed each entity into a low dimensional space if they are mapped together within similar topics. We measure each cell based on the compatibility of the argument

representation and their corresponding semantic topic groups with other cells. This can be more formally represented as:

$$\theta_{r,t} = \sum_k a_{t1,k} e1_{r,k} h_{t1,k} + \sum_k a_{t2,k} e2_{r,k} h_{t2,k} \tag{7}$$

where h_{t1} denotes the vector of topics for argument 1, and h_{t2} denotes vector of topics for argument 2.

Given the fact that using only semantic groups of topics could be noisy for training purposes, we also further augment the topic model with clause-based features. For instance, <(HOLLANDE-FRANCE)> could be learned by <(OBAMA-US)> if they are presented with a similar clause type. This could be presented as:

$$\theta_{r,t} = \sum_k a_{t1,k} e1_{r,k} h_{t1,k} v_{t1} + \sum_k a_{t2,k} e2_{r,k} h_{t2,k} v_{t2} \tag{8}$$

where v_{t1} is the vector of clause type for argument 1, and v_{t2} is the vector of clause type for argument 2.

3.5 Interpolated Models

Each of the above models represents a unique and important aspect of the data that needs to be combined with other models to predict potential relations in the matrix. In practice, combining the introduced models can capture different necessary aspects of the data. For instance, the combined model of Entity and Neighbor can take advantage of selectional preference on argument slot presentation from the Entity model and the weight of the related neighbors from the Neighbor model. We linearly interpolate the models, e.g., the combination of F, N, E and T models can be shown as follows:

$$\theta_{r,t} = F(\theta_{t,r}) + N(\theta_{t,r}) + E(\theta_{t,r}) + T(\theta_{t,r}) \tag{9}$$

3.6 Parameter Estimation

Similar to the F model, relation cells in the matrix model are parameterized through weights and/or latent component vectors. In each model, we predict a relation with a number between 0 and 1. However, the models require negative training data for the learning process. We train the models by ranking the positive cells (observed true facts) with higher score than the negative cells (false facts). The log-likelihood setting could be contrasted with this constraint that primarily requires negative facts to be scored below a defined threshold. Thus, it is possible to calculate the gradient for the weights of cells. We also use log-likelihood as the objective function and employ stochastic gradient descent with a logistic function $\sigma(\theta_{r,t}) = 1/(1 + \exp(-\theta_{r,t}))$ to learn the parameters $x_{r,t} = \sigma(\theta_{r,t})$.

4 Experimentation

4.1 Experimental Setting

In this paper, in order to benchmark our approach, we conducted experiments on the dataset[1] proposed in [1]. The content of this dataset is comprised of reports from New York Times where each sentence has been annotated with relation types with linked entity tags to Freebase. Note that, we do not use the dataset from [15] given the fact that it does not include the original sentences, which prevents us from being able to identify grammatical clauses as required in our approach. We used ClausIE [5] to extract the clause patterns and then check them with entity tuples annotated in each sentence in order to embed them into the matrix. For embedding clause types into the matrix, we use three fundamental clause types, namely SVO, SVC and SVA. The details of these clauses are presented in [5]. Given we only focus on three clause types, if a tuple of entities was extracted with a different clause type, e.g., "Bill has worked for IBM since 2010" that corresponds to the SVOA clause pattern, we check the main entities of the relation's corresponding elements and convert its clause type into one of the three main types of clauses. In this case, SVOA will be converted into SVO because "Bill" represents S and "has worked" denotes V, and "IBM" represents O.

Additionally, for extracting the semantic groups of topics, we generate and estimate topic models based on LDA through Gibbs Sampling using GibbsLDA++[2]. We optimize three important parameters a, b and number of topics T in the LDA. It is based on the topic number and the size of the vocabulary in the document collection, which are $a = 50/T$ and $b = 0.01$, respectively [13]. Then we vary topic sizes between 100, 150, and 200. We evaluate each group of topics and select topic size 150, which shows the best performance for our experiments.

4.2 Experimental Results

We have conducted experiments on both individual models and interpolated models for predicting relations as listed in Tables 1 and 2. We randomly split the dataset for training and testing and applied 10-fold cross validation for all models. We have applied the threshold 0.5 as suggested in [15] for all models that indicate the confidence value to predict a relation. Table 1 shows the detailed performance of each model as well as the combined models in Table 2. As observed in the table, using clause features shows improved performance compared to when models are built without clause information. Using the clause information, we can see the EC model with F-measure of 41.81% is better than the E model with F-measure of 38.77%; N model obtained only 36% in F-measure while NC obtained 39.5% in F-measure. We observe that, N models are lower than the other models due to weak co-occurrence with other relations. The interpolation of N, F, E and T models outperforms the non-interpolated models, indicating the power of selectional preferences learned from data, e.g., F+E+N (being

[1] http://nlp.stanford.edu/software/mimlre-2014-07-17-data.tar.gz.

[2] http://gibbslda.sourceforge.net.

Table 1. Experimental results in individual models.

Models	Precision (%)	Recall (%)	F-measure (%)
E	48.23	32.41	38.77
EC (E with clause)	51.97	37.02	**41.81**
N	44.61	30.18	36.00
NC (N with clause)	48.94	33.11	**39.50**
T	46.79	41.70	44.10
TC (T with clause)	54.71	37.02	**44.16**
F	58.02	39.26	**46.83**

Table 2. Experimental results in interpolated models.

Models	Precision (%)	Recall (%)	F-measure (%)
Baseline [15] (F+E+N)	**79.58**	38.51	51.90
F+E+N+T	51.16	53.30	52.21
EC+NC	72.29	32.51	43.88
TC+NC	64.12	34.98	47.82
EC+TC	59.58	39.67	47.62
F+EC	54.65	42.36	47.69
F+NC	56.24	40.14	46.85
F+TC	53.02	46.87	49.75
NC+EC+TC	57.24	42.36	48.69
F+EC+NC	57.31	49.24	52.96
F+NC+TC	55.01	54.80	54.90
F+EC+NC+TC	60.23	**60.00**	**60.11**

the baseline presented by Reidel et al. [15]) and F+E+N+T models have an F-measure of 51.9% and 52.23%, respectively.

The interpolated models EC+NC, EC+TC, and EC+TC+NC benefit from important aspects of the data from the EC, NC and TC models and take advantage of selectional preference on argument slot presentation from entities and the weight of the related neighbors. EC+NC achieves an F-measure of 43.88%, EC+TC has an F-measure of 47.62% and EC+TC+NC produces an F-measure of 48.69%. Therefore, the interpolated models obtain better results compared to the individual EC, NC, or TC models. We observed that TC employs features based on the presentation of argument slots from entities; and the presentation of argument slots in the TC model results in a much higher number of co-occurrences compared to the EC model. Therefore, the interpolated models with TC achieve better results compared to the interpolated models with EC, e.g., TC+NC yielded 47.82% while EC+NC yielded 43.88%.

The interpolated models with F such as F+TC, F+NC+TC and F+EC+NC+TC have sufficient number of features, which are employed based on PCA components (F model). Therefore, F+TC, F+NC+TC and F+EC+NC+TC achieve better results compared to the interpolated models without F such as TC, NC+TC, and EC+NC+TC. For instance, NC+TC obtains an F-measure of 47.82% while F+NC+TC obtains

54.90%. Finally, the best interpolated model is F+EC+NC+TC which produces the highest result with 60.11% in F-measure when compared to the other models. Our interpolated models, namely F+NC+TC, F+E+N+T and F+EC+NC+TC outperform the baseline (F+E+N) proposed in [15].

Finally, we would like to summarize the impact of our proposed work on performance. As seen in the table, when employing clause types on the baseline (F+E+N vs. F+EC+NC), we see that precision drops; however, recall increases and overall the incorporation of clause type improves F-measure. Also when adding semantic topics to the baseline (F+E+N vs. F+E+N+T), we see a similar trend. The important observation is that once clause types and topic models are added simultaneously (F+EC+NC+TC) that we achieve a significant improvement on recall and a reasonable precision performance, leading to much higher F-measure. This shows that clause types and semantic topics can help identify a higher number of relevant relations and hence increase retrieval rates and also maintain acceptable precision.

Table 3 shows several specific relation types in our models. We show the top-5 relation types that have the best F-measure scores in the top-6 best performing models such as the F model (the best individual model), EC+NC+TC (the best interpolated model without F), F+E+N and F+E+N+T (the two best models without clause types), F+TC+NC and F+EC+NC+TC (the best interpolated models with clause types). These relations take advantage of selectional preference in the training process due to their co-occurrence and/or clause type similarity with other relations. For instance, similar entities co-occur multiple times in relations such as "org/country_of_headquarters", "org/city_of_headquarters" or "per/country_of_birth"; therefore, making the cells for these relationships highly similar and related to each other. Hence, the models will take advantage of such similarity in the training process in order to learn latent relations between the relationships. Beside entity co-occurrences, some relations appear only within a specific clause type. e.g., "person/founded" is often seen in the "SVC" clause type while "org/member_of" is observed as the "SVO" clause type; hence, the application of the clause type information can significantly help find similarity or relationship between these relations and the others in the matrix and lead to reduced noise in the training process.

Table 3. F-measures of top-5 relation types in the best six models.

	F	EC+NC+TC	F+E+N	F+E+N+T	F+TC+NC	F+EC+NC+TC
org/country_of_headquarters	60.97	67.22	70.49	76.42	69.16	77.39
person/founded	61.53	62.63	43.68	72.50	72.99	75.84
org/city_of_headquarters	25.84	66.66	63.93	75.60	75.00	76.52
per/country_of_birth	56.92	69.31	79.63	73.54	72.25	76.88
org/member_of	58.96	79.73	78.70	71.59	76.72	70.22

4.3 Discussions

In terms of the performance of the individual models, we observe that the E and T models outperform the N model. The E and T models employ the presentation of

argument slots while N employs co-occurrence with neighbors. The N model might face situations where only a few co-occurrences with other neighbor relations are observed that can cause weak evidence in the training process for learning hidden relations. However, in the T and E models where their argument slots are presented in high dimensions, this could increase the number of desirable co-occurrences. These models take advantage of selectional preference in the training process due to the exploitation of co-occurrence information with other relations. Most of the models have increasing performance when applying clause type features because the clause type information can reduce noise in the training process.

Interpolated models benefit from the advantages of each individual model. Thus, most of the interpolated models achieve better results compared to separate models. Comparing our best models (F+NC+TC) and (F+EC+NC+TC) with Reidel et al.'s model (F+E+N) as a baseline, the results reveal that we obtained 55.01% of precision and 54.80% of recall in F+NC+TC, and 60.23% of precision and 60% of recall in F+EC +NC+TC while Reidel et al. achieved 79.58% of precision and 38.51% of recall. Applying semantic topics to the models could reduce precision but increase recall significantly when compared to the baseline. Baseline+Topic model (F+E+N+T) achieves 51.16% of precision and 53.30% of recall. Our model obtained an improvement in recall when compared to the baseline. However, our models also show lower precision because applying topic-based features in our models will lead to an increasingly higher number of hidden relations for prediction compared to the baseline. This can cause a lower precision in our model even when our model predicts more hidden relations compared to Reidel et al.'s model. Finally, based on the F-measure metric, our models show up to 8% improvement in comparison to the baseline model.

Now, let us look at some of the major causes of error in our proposed models. There are some factors, which can affect the results. First, some relation types show missing evidence for training that cause low accuracy when predicting latent relations. For example, the relation "per:cause_of_death" has only been observed very few times with other relations in the matrix. Consequently, after the training process, the trained models do not have enough knowledge to predict such infrequent relations. Second, there are incorrect linked entities that cause noise in the matrix. We found that some tuples of entities, which are linked to entities from Freebase, are not accurately placed in the correct tuple or relation in the dataset. For example, "Obama, who is the President of US, has visited Canada" has been annotated with the tuple of entities <(OBAMA-CANADA)> with relation "person: employee". Therefore, such a tuple in the training set will introduce noise, which can lead to issues when predicting relations. Finally, ambiguous tuples of entities might occur in the dataset, e.g., tuple of entities (<WASHINGTON-US>) areseen in several relations such as "org:country_of_headquarters", "per:countries_of_residence", and "per:origin" because "WASHINGTON" could refer to a city in some cases, or a person in other cases that leads to noise in the training processes. As a result, this will have a negative effect on performance when predicting hidden relations.

5 Concluding Remarks

In this paper, we presented several matrix models with feature enrichment for predicting potential relations. We have exploited universal schemas that are formed as a collection of patterns from OIE and relation schemas from pre-existing datasets to build a matrix model in order to use matrix factorization and collaborative filtering to predict hidden relations. While previous systems have trained relations only for entities, we further exploited advanced features such as clause types and semantic topics for predicting hidden relations. Particularly, we exploited clause-based features extracted from OIE combined with semantic groups of topics, which are used as important characteristics for predicting potential relations. In our experiments, the results reveal that our proposed models achieve better results compared to the state of the art, which demonstrates the efficiency of our proposed approach.

References

1. Angeli, G., Tibshirani, J., Wu, J., Manning, C.D.: Combining distant and partial supervision for relation extraction. In: EMNLP 2014 (2014)
2. Blei, D., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Bollegala, D., Matsuo, Y., Ishizuka, Y.: Relational duality: unsupervised extraction of semantic relations between entities on the web. In: WWW 2010 (2010)
4. Collins, M., Dasgupta, S., Schapire, R.S.: A generalization of principal component analysis to the exponential family. In: NIPS 2001 (2001)
5. Corro, L.D., Gemulla, R.: ClausIE: clause-based open information extraction. In: WWW 2013 (2013)
6. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: EMNLP 2011 (2011)
7. Greenwood, M.A., Stevenson, M.: Improving semi-supervised acquisition of relation extraction patterns. In: IEBD 2006 (2006)
8. Kambhatla, N.: Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In: ACL 2004 (2004)
9. Kemp, C., Tenenbaum, J.B., Griffiths, T.L.: Learning systems of concepts with an infinite relational model. In: AAAI 2006 (2006)
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: KDD 2009 (2009)
11. Mausam, Schmitz, M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: EMNLP 2012 (2012)
12. Pantel, P., Pennacchiotti, M.: Espresso: leveraging generic patterns for automatically harvesting semantic relations. In: COLING 2006 (2006)
13. Phan, X.H., Nguyen, C.T., Le, D.T., Nguyen, L.M., Horiguchi, S., Ha, Q.T.: A hidden topic-based framework toward building applications with short web documents. IEEE Trans. Knowl. Data Eng. **23**, 961–976 (2011)
14. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bayesian personalized ranking from implicit feedback. In: Proceedings of UAI 2009 (2009)

15. Riedel, S., Yao, L., McCallum, A., Marlin, M.: Relation extraction with matrix factorization and universal schemas. In: NAACL 2013 (2013)
16. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: EMNLP-CoNLL 2012 (2012)
17. Takamatsu, S., Sato, I., Nakagawa, H.: Probabilistic matrix factorization leveraging contexts for unsupervised relation discovery. In: PAKDD 2011 (2011)
18. Vo, D.T., Bagheri, E.: Self-training on refined clause patterns for relation extraction. Inf. Process. Manage. (2017). doi:10.1016/j.ipm.2017.02.009
19. Vo, D.T., Bagheri, E.: Open information extraction. Encycl. Semant. Comput. Robot. Intell. **1**(1) (2017). doi:10.1142/S2425038416300032
20. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: ACL 2010
21. Zhou, G., Qian, L., Fan, J.: Tree kernel based semantic relation extraction with rich syntactic and semantic information. Inf. Sci. **180**, 1313–1325 (2010)