Information Processing and Management 56 (2019) 424–444

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Feature-enriched matrix factorization for relation extraction

Duc-Thuan Vo, Ebrahim Bagheri*

Laboratory for Systems, Software and Semantics (LS3), Department of Electrical and Computer Engineering, Ryerson University, Toronto, Canada

ARTICLE INFO

Keywords:
Open Information Extraction
Relation Extraction
Matrix Models
Matrix Factorization

ABSTRACT

Relation extraction aims at finding meaningful relationships between two named entities from within unstructured textual content. In this paper, we define the problem of information extraction as a matrix completion problem where we employ the notion of universal schemas formed as a collection of patterns derived from open information extraction systems as well as additional features derived from grammatical clause patterns and statistical topic models. One of the challenges with earlier work that employ matrix completion methods is that such approaches require a sufficient number of observed relation instances to be able to make predictions. However, in practice there is often insufficient number of explicit evidence supporting each relation type that could be used within the matrix model. Hence, existing work suffer from a low recall. In our work, we extend the work in the state of the art by proposing novel ways of integrating two sets of features, i.e., topic models and grammatical clause structures, for alleviating the low recall problem. More specifically, we propose that it is possible to (1) employ grammatical clause information from textual sentences to serve as an implicit indication of relation type and argument similarity. The basis for this is that it is likely that similar relation types and arguments are observed within similar grammatical structures, and (2) benefit from statistical topic models to determine similarity between relation types and arguments. We employ statistical topic models to determine relation type and argument similarity based on their co-occurrence within the same topics. We have performed extensive experiments based on both gold standard and silver standard datasets. The experiments show that our approach has been able to address the low recall problem in existing methods, by showing an improvement of 21% on recall and 8% on f-measure over the state of the art baseline.

1. Introduction

Relation Extraction (RE) has emerged as one of the mainstream tasks within the information retrieval and natural language processing communities that aims at systematically discovering relations between two arguments from a textual corpus. To this end, many of the existing approaches use a predefined, finite and fixed schema of relation types in the context of supervised (Bunescu & Mooney, 2005; Kambhatla 2004; Zhou & Zhang 2007; Zhou, Qian, & Fan, 2010; Barrio and Gravano, 2017), semi-supervised (Agichtein & Gravano, 2000; Xu, Uszkoreit, & Li, 2007; Xu, Uszkoreit, Krause, & Hong Li, 2010; Zhang et al., 2015) and unsupervised learning techniques (Etzioni et al., 2005; Rosenfeld and Feldman, 2007; Turney, 2008; Akbik et al, 2012; Oramasa, S., Espinosa-Ankeb, L., Sordoc, M., Saggionb, H., & Serraa, H. 2016; Vlachidis and Tudhope, 2016; Yao, Riedel, & McCallum, 2012). The main strategy used in supervised methods is to generate linguistic features based on syntactic, dependency, or shallow semantic structures of text. Based on these features, the models are then trained to identify pairs of entities that might be related through some relation,

* Corresponding author.
E-mail addresses: thuanvd@ryerson.ca (D.-T. Vo), ebrahim.bagheri@gmail.com, bagheri@ryerson.ca (E. Bagheri).

https://doi.org/10.1016/j.ipm.2018.10.011
Received 18 April 2018; Received in revised form 20 September 2018; Accepted 12 October 2018
Available online 08 January 2019
0306-4573/

and then to classify them based on a predefined set of relation types. In contrast, semi-supervised techniques employ an initial seed set of, often manually labeled, relations, which are used to extract patterns that can extract additional relations from text. The newly extracted relations based on the learnt patterns are then iteratively used to update the initial seed set and the process is repeated until certain stopping criterion is met. These approaches require corpora that include sufficient example relation instances that might be time consuming to prepare.

Unsupervised techniques, often referred to as Open Information Extraction (OpenIE), (Etzioni et al., 2011; Banko, Cafarella, Soderland, Broadhead, & Etzioni, 2007; Corro & Gemulla, 2013; Wu & Weld, 2010; Mausam, Schmitz, Bart, & Soderland, 2012; Vo & Bagheri, 2017) focus on extracting relations with minimal domain-dependent background knowledge and the least amount of annotated training data. In this context, *distant supervision* (Angeli, Tibshirani, Wu, & Manning, 2014; Riedel, Yao, McCallum, & Marlin, 2013; Surdeanu, Tibshirani, Nallapati, & Manning, 2012) techniques exploit information from external knowledge bases, such as Freebase, in order to perform large-scale relation extraction from text. Distant supervision approaches often avoid dependence on training samples by using natural language grammatical structures or semantic word senses to define and model *universal schemas*. To this end, Riedel et al. (2013) have presented a matrix factorization model based on universal schemas for predicting relations. These authors presented a set of models that learn lower dimensional manifolds for tuples of entities and relations with a set of weights in order to capture and model relationships between relation types. In this context, the output of the first generation of OpenIE systems, such as TextRunner (Banko et al., 2007) and WOE (Wu & Weld, 2010), are used for building the universal schemas.

While approaches based on universal schemas have shown reasonable performance, their limitation is in that they are trained for relationships between specific entity tuples and relation types, and therefore, are limited when an insufficient number of explicit evidence is present for each relation type for specific entity tuples. For instance, the relation (“Hawking”, “professor-of”, “Cambridge Univ.”) could not help us infer a similar yet unobserved relation (“Wiles”, “professor-of”, “Oxford Univ.”) due to the differences of the entity tuples, i.e., (“Hawking”, “Cambridge Univ.”) and (“Wiles”, “Oxford Univ.”). Furthermore, such systems also fall short in predicting other relation types between the same entity tuples, e.g., (“Obama”, “is-president-of”, “US”) and (“Obama”, “has-returned-to”, “US”), which are between the same entity tuples but with different relation types. In addition, these approaches learn linear chain models based on unlexicalized features such as part of speech or shallow tags to label the intermediate words between pairs of potential arguments for identifying extractable relations. However, they do not employ deeper linguistic analysis on the grammatical structure of a sentence such as clause level analysis and therefore, they might suffer from problems such as extracting incoherent and uninformative relations. So it is possible to summarize the limitations of the current work on universal schemas as follows:

1. The matrix models built based on the universal schemas are trained to predict relation types between specifically observed entity tuples; hence, same relation types cannot be predicted for other different yet semantically-related entity tuples;
2. Similarly, the relation type between entity tuples is predicted based on a matrix factorization model where the participation of entities in other already observed relations determines an unobserved relation and as such other semantically-relevant yet unobserved relations cannot be determined;
3. Universal schemas are primarily built based on part of speech tags and shallow analysis of the textual content; however, deeper linguistic analysis such as considering entity and relation context within the sentence structure, e.g., sentence clause structure, is not yet considered.

1.1. Research objectives and contributions

In light of the above limitations, while existing work based on universal schemas and matrix models have a reasonable precision in retrieving correct relations, they do not perform as well on the recall measure. In other words, given matrix models, such as matrix factorization, require substantial amount of evidence to draw conclusions, and also the fact that there are often very limited set of explicit evidence supporting relation instances, these models fall short in showing good recall performance. In order to address this challenge and improve the recall of existing work, we propose that additional features that can serve as implicit indicators of relation instances need to be introduced. To this end, we introduce and exploit new features based on grammatical clause types and statistical topic models to enrich universal schemas used in the matrix factorization model for predicting new relation instances. Particularly, we exploit clause types and topic models to predict relations regardless of whether they were explicitly observed at training time with direct or indirect access. This allows us to make predictions on relation types that have not been explicitly observed in the training corpus that can hence lead to improved recall performance.

Our work uses the concept of universal schemas from Riedel et al. (2013) in order to convert the knowledge base information combined with OpenIE patterns into a binary matrix representation where entity tuples form its rows and relations are represented as columns. More concretely, the contributions of our work can be enumerated as follows:

1. We propose numerous matrix models with fully enriched features such as word context, selectional preference, clause types and statistical topic models and employ matrix factorization with direct/indirect references for predicting specific relations between entities. We show that the consideration of sentence clause types as well as information from statistical topic models enables us to identify and determine semantically-relevant yet explicitly unobserved relations between entity tuples, as mentioned above;
2. We employ and evaluate the impact of four state-of-the-art OpenIE systems used for constructing and populating the initial matrix models that represent the relations between entity tuples and relation types and show how the characteristics and performance of these systems impact the outcome of our proposed approach.
3. We systematically evaluate our proposed features in isolation and in tandem within the matrix factorization model and study their

impact for identifying relations between entity tuples. We compare our work with the state of the art based on the widely used gold standard by Angeli et al. (2014), which consists of 40 different relation types and over 22,000 relation instances. We also use two silver standard corpora collected from Wikipedia and New York Times to perform additional experiments.

The rest of this paper is organized as follows. Section 2 presents background literature on relation extraction. An overview of the proposed approach is presented in Section 3 where the description of the features employed for relation extraction is shown. Section 4 presents the formal description for computing and incorporating these features into a matrix-based model. This is followed by an in-depth discussion of experimental results in Section 5 where the results are compared to the state-of-the-art and the impacts of different OpenIE systems on performance are studied. Section 6 finalizes the paper with conclusions and future work.

2. Related work

There have been several research work that focus on building universal schemas for identifying relations using matrix factorization (Riedel et al., 2013; Yao, 2015) as well as collaborative filtering (Koren , 2009; Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2009). In this section, we will cover a broader literature on relation extraction beyond the work that is immediately related to our work in this paper.

2.1. General relation extraction

Relation extraction has become a fast evolving research domain within information extraction literature that aims to discover and represent structured relations between segments of text. Several techniques have been proposed for this task such as supervised methods (Abacha and Zweigenbaum, 2016; Singhal, Simmons, & Lu, 2016; Zheng et al., 2017;Zhou and Zhang, 2007; Zhou et al., 2010), unsupervised methods (Etzioni et al., 2005; Rosenfeld et al., 2007; Turney, 2008; Akbik et al, 2012; Oramasa et al., 2016; Vlachidis & Tudhope, 2016; Yao et al., 2012) and semi-supervised bootstrapping methods (Pantel et al., 2006; Xu et al., 2007; Zhang et al., 2015). Supervised methods heavily rely on hand-crafted labeled datasets for training models that learn patterns to identify instances of certain relation types (Abacha et al., 2016; Singhal et al., 2016; Zhou et al., 2007; Zhou et al., 2010). In this context, Zhou and Zhang (2007) and Zhou et al. (2010)) used linguistic pattern learning for relation extraction by exploiting syntactic features, dependency features, and shallow semantic structures of text. These systems were trained to identify pairs of entities, which were classified based on a predefined set of relation types. Unsupervised and semi-supervised bootstrapping methods (Turney, 2008; Akbik et al, 2012; Oramasa et al., 2016, Nguyen, Theobald, & Weikum, 2017; Ryu, Jang, & Kim, 2015; Vlachidis & Tudhope, 2016; Zhang et al., 2015) are often based on some heuristic rules or clustering techniques that work over a large unlabeled corpus. For instance, Akbik et al. (2012) and Yao et al. (2012) used the k-mean clustering algorithm and cosine similarity to build a vector space model to cluster entity pairs by exploiting syntactic and dependency parsing information. In the context of heuristic rule-based approaches, Ryu et al. (2015) have presented a system for open question answering based on information derived from Wikipedia's article content, structure, infoboxes, categories, and definitions, which can be used for general domains. Also, Vlachidis and Tudhope (2016) extracted relation patterns by defining a set of rules based on syntactic analysis. Zhang et al. (2015) presented a semantic bootstrapping with bottom-up kernel method that uses semantic patterns and applies matching algorithms for relation extraction. Further, Oramasa, Espinosa-Ankeb, Sordoc, Saggionb, and Serraa (2016) defined rules to extract potential relations between entities, which have been discovered by traversing the dependency tree by exploiting syntactic and semantic information.

2.2. Open information extraction (OpenIE)

Banko et al. (2007) have introduced the pioneering OpenIE system, known as TextRunner. Several OpenIE systems have since been developed in recent years to build upon TextRunner. Some existing works use shallow syntactic representation (Banko et al., 2007; Fader, Soderland, & Etzioni, 2011; Wu & Weld, 2010) while other works use dependency parsing outputs in the form of verbs or verbal phrases and their arguments (Corro & Gemulla, 2013; Mausam et al., 2012; Nebot and Berlanga, 2014; Wu & Weld, 2010). OpenIE approaches could be viewed as having two generations. The first generation of OpenIE systems makes use of syntactic parsing. TextRunner (Banko et al., 2007) and ReVerb (Fader et al., 2011) are two prominent systems from the first generation that use training data and syntactic analysis. TextRunner trains a Naïve Bayes classifier in an offline phase and applies it for the efficient extraction of propositions in the online phase. In contrast, instead of extracting entities first, Reverb extracts verbal relation sequences based on a set of POS patterns. Then entities are identified around the relation sequences, so the system only extracts relation tokens between two entities. The second generation of OpenIE systems focuses on the use of dependency parsing for information extraction. WOE^{parse} (Wu & Weld, 2010) uses automatically generated training data to learn extraction patterns based on dependency parsing. Mausam et al. (2012) present OLLIE that uses hand-labeled data to create a training set, which includes millions of relations extracted by ReVerb (Fader et al., 2011). OLLIE learns relation patterns from the dependency path and lexicon information such that the relations that match the identified patterns will be extracted. Zouaq, Gagnon, and Jean-Louis (2017) have assessed the role of open relation extractors, which exploit OpenIE in the context of the Semantic Web and Linked Data. More recent OpenIE systems, such as ClausIE (Corro & Gemulla, 2013) and LS3RyIE (Vo & Bagheri, 2018), use dependency parsing and a small set of domain-independent lexica without any post-processing or training data. These systems exploit linguistic knowledge about the grammar of the English language to first detect clauses in an input sentence and to subsequently identify the type of each clause according to the grammatical function of its constituents. Therefore, these approaches are able to generate high-precision extractions and can be flexibly

customized to the underlying application domain.

2.3. Matrix factorization-based methods

The objective of matrix factorization and collaborative filtering methods in relation extraction is to predict hidden relations that might not have been explicitly observed. Kemp, Tenenbaum, and Griffiths (2006) 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, Matsuo, & Ishizuka (2010) try to explore clusters of entity pairs and patterns jointly as latent relations by employing co-clustering. Takamatsu, Sato, and Nakagawa (2011) use probabilistic matrix factorization with Singular Value Decomposition to reduce dimensions to discover relations. Kolda & Bader (2009) and Kang *et al.* (2012) employ the tensor product of three vectors, i.e., the vectors for two entities and the vector for the relation for decomposing an entity-entity-relation matrix. These authors employed vector of user-query-page for web recommendation and page-page-anchor for web links. Bordes *et al.*, (2013) and Weston *et al.* (2013) propose a method to determine the first entity vector and its relation vector which leads to the creation of a link to the second entity vector. The goal is to optimize the distance between the second entity vector and the association of the first entity and relation vectors. Riedel et al. (2013) and Yao (2015) use matrix factorization and collaborative filtering by including surface patterns in a universal schema and a ranking objective function to learn latent vectors for tuples of entities and relations. These authors represent each relation as a vector instead of a matrix. Representing each entity as a vector breaks the interaction between two entities. In their systems, the authors use surface patterns extracted from existing OpenIE systems and predict the hidden relations through matrix completion. Similar to Riedel et al. (2013) and Yao (2015), in this study, we employ the notion of universal schemas that is formed as a collection of patterns derived from OpenIE systems as well as from relation schemas of pre-existing knowledge bases. While previous systems have trained relations only for entities, we exploit advanced features from relation characteristics such as clause types and topic models for predicting new relation instances. Our work could naturally predict any tuple of entities and relations regardless of whether they were explicitly observed at training time with direct or indirect access in their provenance.

2.4. Distant supervision

The core idea of distant supervision is to learn a classifier based on a set of weakly labeled corpora that are often annotated using some heuristics. In the area of relation extraction, the work by Mintz, Bills, Snow, and Jurafsky (2009) is among the pioneering works that consider the application of distant supervision techniques. In their work as well as other closely related work such as Surdeanu et al. (2012) and in order to curate a weakly labeled corpus, the authors use the Freebase knowledge base whereby for each pair of entities that are related to each other using some Freebase relation, they will identify sentences in their corpus where these entities have been seen together. This way they are able to extract features that can help them train a classifier for relation extraction. There have been works by Takamatsu *et al.* (2012), Min *et al.* (2013) and Liu, Wang, Chang, and Sui (2017) among others that propose methods to identify low confidence labels that can be removed or ignored. From a different perspective and in order to augment the work in distant supervision, Riedel *et al.* (2010) argue that many of the errors produced by relation extraction techniques are due to the generous interpretation of sentence relevance. In other words, if two entities were related to each other through a Freebase relation, any sentences containing these two entities would be considered related and labeled as such. In Surdeanu et al. (2012), a heuristic method is employed to generate training relations by mapping pairs of mentioned entities in a sentence to corresponding entities in a knowledge base (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. The idea of universal schemas (Riedel et al., 2013) employs the notion of distant supervision by using a KB 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 (Riedel et al., 2013) 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. Overview of our proposed approach

The main premise of the work in this paper is that existing OpenIE systems are able to automatically extract a set of relatively stable relations from a textual corpus based on some heuristic patterns such as analysis of grammatical structure of the sentence in the form of dependency or syntactic parsing or part of speech tagging. While OpenIE systems have shown acceptable performance, they may not be able to extract all possible relations from the text especially in cases when the arguments of the relation have not been explicitly observed within the contexts or forms expected by the OpenIE system. However, it could be possible to capitalize on the relations that have been extracted by OpenIE systems with specific relation types to explore the possibility of inferring other unobserved relations between entities. One of the systematic ways of achieving this objective would be to view this task as a matrix completion process whereby the rows of the matrix are entity pairs and the columns are relation types. Such a matrix could be partially filled based on the relations derived by OpenIE systems and other potential relations between other entity pairs and relation types could be identified by the matrix completion process. Fig. 1 shows an overview of this process where a collection of documents are fed into OpenIE systems whose output relations are then used to build the matrix representation in which the explicit relations extracted by the OpenIE system would be denoted by cells consisting of 1 between pairs of entities and relation types.

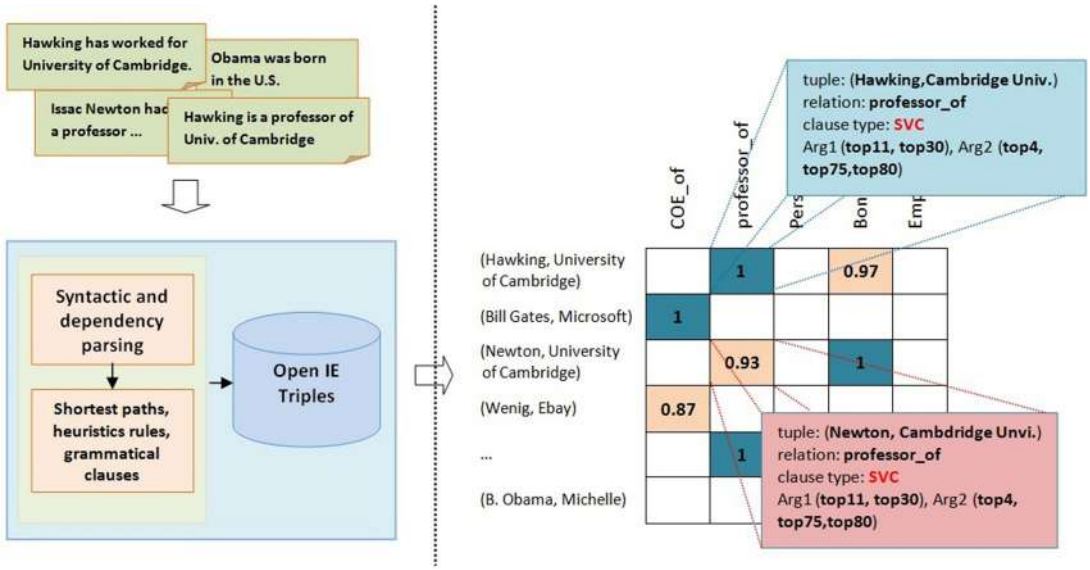


Fig. 1. Overview of the Proposed Approach.

In our work and inspired by Riedel et al. (2013), we find unobserved yet valid relations by using probabilistic matrix factorization to efficiently estimate vector embeddings for both entities and relation types through stochastic gradient descent optimization. The probability of assigning a relation type to an entity pair is determined by the dot-product of the corresponding embeddings, mapped through a logistic function. While matrix factorization will project the relation matrix into two matrices that exhibit some latent structure in the identified relations, which we refer to as the *latent feature,* we additionally define four other types of features to further enhance the performance of the matrix factorization method, namely *word context feature, selectional preference feature, statistical topic model-based feature* and *clause-type feature.* It is possible to perform the matrix completion task based on each individual feature as well as the integration (interpolation) of these features. The overview of each of the features is provided in the following.

a. Word Context Feature: The first type of feature that we consider is the context in which a pair of entities or the relation type of that relation happen. We define *context* to be the set of words observed before or after an item of interest such as the entities in a relation instance or the relation type. The reason it is important to consider word context as a feature in our work is because the objective of our work is to identify unobserved relations and hence if an entity pair is frequently observed within the same context as another entity pair for which we already explicitly observed some relation type, then it would be possible to deduce that such relation might also exist between the entities of the first pair. Poon & Domingos (2008) and Koren (2009) have argued that those words, which occur in similar contexts tend to have similar meanings, which allows us to argue that it would be possible to assume that the entities of two similar entity pairs would be related to each other with similar relation types. For instance, “Professor” and “Principal Investigator” are often seen in similar relation instances shown in Fig. 2. Therefore, it would be possible to probabilistically infer that two entities that are related to each other through the “Professor” relation could also be related to each other through the “Principal Investigator” relationship as well. In the matrix model, the word context feature could be used to define a neighborhood relationship between relations based on the similarity of their contexts.
b. Selectional Preference Feature: The motivation behind this feature rests on the understanding that only specific types of entities can be used to fill in the relation argument roles for any given relation type. In the context of relations, selectional preference can refer to the constraints that relation types impose on their arguments. The application of selectional preference allows one to not only refine incorrectly identified relation instances, but also identify entity pairs that satisfy the constraints of a relation type and hence can act as a candidate for serving as a relation instance. As an example, consider the relation type “Professor” in Fig. 2 that requires its arguments to be entities of type *scholar* and *academic institution.* Therefore, any pair of entities of type scholar and academic institution would be intuitively a candidate for a relation instance of type “Professor”. It is clear that this feature will result in the extraction of many false positive relations; however, it can be envisioned that when used in conjunction with other features such as the word context feature, many irrelevant relation candidates will be ruled out. Therefore, the selectional preference feature can be seen as an effective tool for increasing the recall of the relation extraction model.
c. Statistical Topic Model-based Feature: The third type of feature that we consider is in essence a semantic extension of the word context feature. While neighborhood is defined based on the similarity of entity pair and relation type contexts in the word context feature, in the statistical topic model-based feature, neighborhood is defined based on the membership of entity pairs and relation types to the same topics derived by a topic model. In other words, two entities can be considered to be semantically related to each other if they belong to the same topics. As such, it is possible to probabilistically assume that entities within the same topic as the entities participating in a relation instance could participate in a similar relation type. The underlying reason for this is that topic

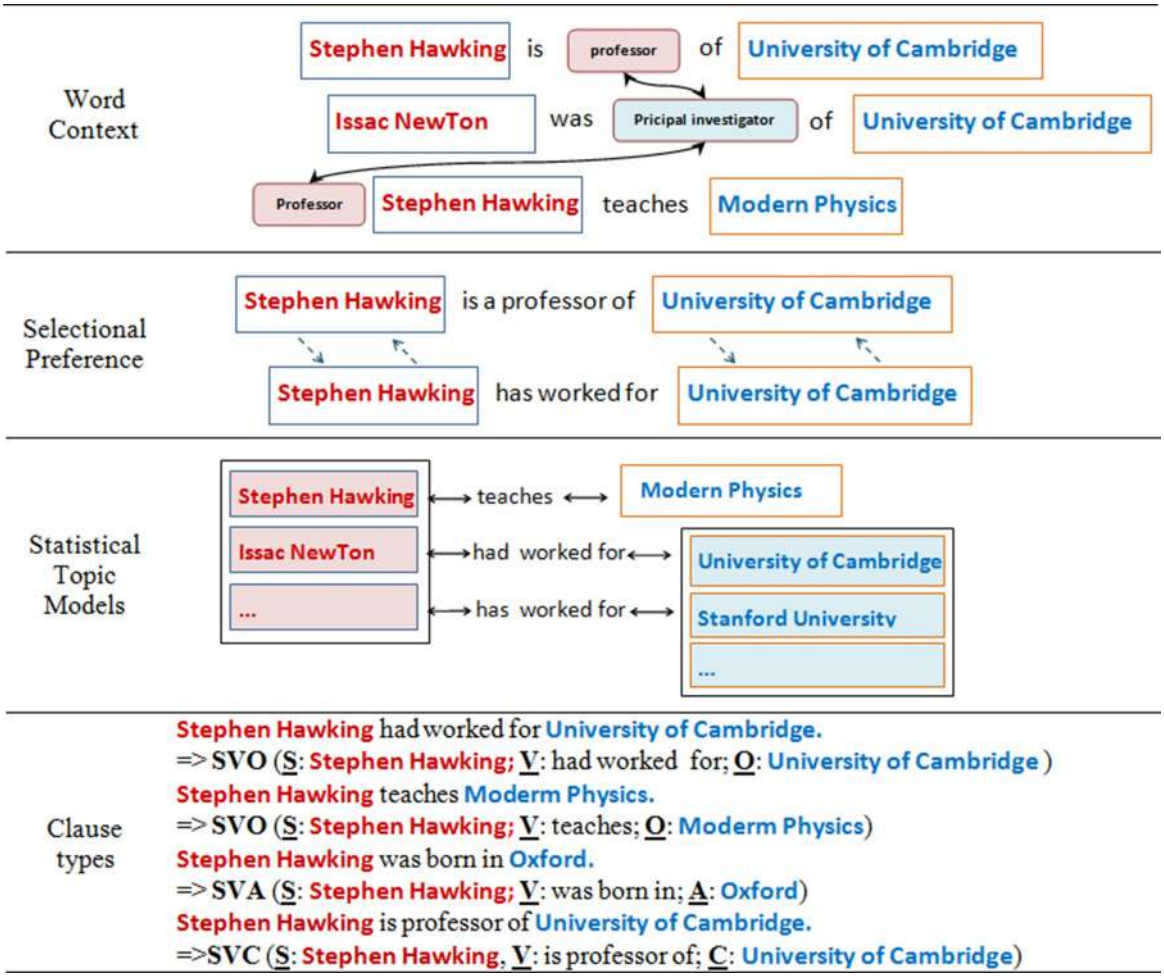


Fig. 2. The four feature types employed in our work.

modeling methods provide a probabilistic framework based on term co-occurrences within the documents of a given corpus. Topic models produce probability distributions over words in topics and documents in topics, which can assist in identifying highly similar terms based on their co-occurrence in similar topics. In other words, topics derived by topic models could be seen as the *semantic clustering* of terms based on which a neighborhood model could be defined. For instance, as shown in Fig. 2, the neighborhood model would deduce that ("Isaac Newton", "teaches", "Physics") given an explicitly observed relation in the corpus ("Stephen Hawking", "teaches", "Physics") and the fact that "Isaac Newton" and "Stephen Hawking" were observed in the same topic derived by the statistical topic model on the document collection. In the matrix model, the topic model-based information is used to define a neighborhood model.

d. Clause-Type Feature: The previous three features are primarily included to benefit from some form of context similarity to infer a neighborhood model for determining unobserved yet reasonable new relations. This feature, however, defines context based on the grammatical role that entity pairs or relation types play within a given sentence. Considering the fact that sentence *clause structure* has already been shown to be a suitable grammatical structure for identifying relations within a sentence (Corro & Gemulla, 2013; Quirk et al., 1985), we employ clause types and clause components in this feature. Technically speaking, a clause can consist of different components including subject (S), verb (V), indirect object (O), direct object (O), complement (C), and/or one or more adverbials (A). As demonstrated in Corro and Gemulla (2013) and Quirk et al. (1985), a clause can be categorized into different types based on its constituent components. Given a clause in a relation, it is possible to determine its type of relation via relation presentation of S, V, O and C such as SVO, SVC, and SVOO depicted in Table 1. Given the clause type of the relation instance, one possibility for generating new relation instances is to use entity pairs that have appeared in similar clause patterns within the corpus to form a new relation instance of that relation type. Similar to the selectional preference feature, the clause-type feature can also lead to increased recall at the expense of a higher false positive rate, which can be mitigated when integrated with the other features.

In the following section, we formally introduce how each of the proposed features is implemented and integrated when required.

Table 1
Sample clause types (Corro & Gemulla, 2013; Quirk *et al.*, 1986); S: Subject, V: Verb, A: Adverbial, C: Complement, O: Object.

Clause types	Sentences	Patterns	Derived clauses
SV	Albert Einstein died in Princeton in 1955.	SV	(Albert Einstein, died)
		SVA	(Albert Einstein, died in, Princeton)
		SVA	(Albert Einstein, died in, 1955)
		SVAA	(Albert Einstein, died in, 1955, [in] Princeton)
SVA	Albert Einstein remained in Princeton until his death.	SVA	(Albert Einstein, remained in, Princeton)
		SVAA	(Albert Einstein, remained in, Princeton, until his death)
SVC	Albert Einstein is a scientist in the 20th century.	SVC	(Albert Einstein, is, a scientist)
		SVCA	(Albert Einstein, is, a scientist, in the 20 the century)
SVO	Albert Einstein has won the Nobel Prize in 1921.	SVO	(Albert Einstein, has won, the Nobel Prize)
		SVOA	(Albert Einstein, has won, the Nobel Prize, in 1921)
SVOO	RSAS gave Albert Einstein the Nobel Prize.	SVOO	(RSAS, gave, Albert Einstein, the Nobel Prize)
SVOA	The doorman showed Albert Einstein to his office.	SVOA	(The doorman, showed, Albert Einstein, to his office)
SVOC	Albert Einstein declared the meeting open.	SVOC	(Albert Einstein, declared, the meeting, open)

4. Formalization of the proposed approach

The objective of our work is to predict the hidden relations by completing the schema in the matrix built from surface patterns and fixed relations. Using the same notation as Riedel et al. (2013) and Yao (2015), we use T and R to correspond to entity tuples and relations. Given a relation $a_r \in R$ and a tuple $e_t \in T$, the objective of our work is to derive a fact about a relation a_r and a tuple of two entities e_t. 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 a_r is true for the tuple e_t, and 0 when relation a_r is false for e_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 based on the features introduced in the previous section to address the task as follows.

4.1. Model-based on the latent feature (F model)

The model based on the latent feature derives the latent relations based on the matrix factorization approach, hence we refer to it as the *F model*, where we denote each relation by a_r and each tuple of entities as e_t. We measure compatibility between relation a_r and tuple e_t as the dot product of two latent feature representations of size k. Thus we have:

$$\theta^F_{r,t} = \sum_k a_{r,k} e_{t,k} \tag{1}$$

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 (Collins, Dasgupta, & Schapire, 2001). Thus, a model with the matrix $\Theta = (\theta^F_{r,t})$ of natural parameters is defined as the low rank factorization AE. To estimate the values in PCA, we have:

$$\sigma(\theta^F_{r,t}) = \sigma\left(\sum_k a_{r,k} e_{t,k}\right) \tag{2}$$

Here, we are applying a logistic function $\sigma(\theta^F_{r,t}) = 1/(1 + \exp(-\theta^F_{r,t}))$ (Collins et al., 2001; Koren et al., 2009; Riedel et al., 2013) to model a binary cell in the matrix. Each cell is drawn from a Bernoulli distribution with natural parameter $\theta^F_{r,t}$. Following Yao (2015), adding a prior for the parameters, the gradient with respect to $\theta^F_{r,t}$ is as follows:

$$(1 - (\theta^F_{r,t}))\frac{\partial}{\partial_{\theta^F_{r,t}}}\theta^F_{r,t} - \lambda_{\theta^F_{r,t}}\theta^F_{r,t} \tag{3}$$

Applying gradients of $\theta^F_{r,t}$ with regards to the parameters $a_{r,k}$ and $e_{t,k}$, we have:

$$\frac{\partial}{\partial_{e_{t,k}}}\theta^F_{r,t} = a_{r,k} \tag{4}$$

$$\frac{\partial}{\partial_{a_{r,k}}}\theta^F_{r,t} = e_{t,k} \tag{5}$$

We maximize the log-likelihood of the observed cells under a probabilistic model to learn low dimensional representations as:

$$\max \sum_r \sum_{\neg r} \log[\sigma(\theta^F_{r,t} - \theta^F_{\neg r,t})] \tag{6}$$

The representations a_r and e_t can be found by maximizing the log-likelihood using stochastic gradient descent. In this model, the existence of a certain relation type between an entity pair is estimated based on the value determined for the corresponding matrix

cell through the optimization problem. As explained later in our experiments, cells with scores above a predefined threshold are considered as true.

4.2. Model-based on the word context feature (N model)

Based on the word context feature and within our matrix formalization, a relation in a column could be neighbor to some other co-occurring relation (Koren, 2009) (hence called the *N model*). For example, the relations "Professor-of" and "Investigator-of" are often seen in similar contexts. Therefore, the word context feature 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 "Researcher", "Dr." or "Professor" often appears in tuples of entities in relations such as "Professor-of" and "Investigator-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:

$$\theta^{N}_{r,t} = \sum_{k} w_k f_k(a_{r'_{att}}; a_{r_{att}}) \tag{7}$$

where w_k is the weight of the association between $a_{r'}$ and a_r; $f_k(a_{r'}; a_r)$ defines a conjunctive feature between relation a_r and neighboring relation $a_{r'}$, e.g., $a_{r_{att}} = <$ att:"professor" $>$ and $a_{r'_{att}} = <$ att:"researcher" $>$; k is the number of relations that have the exact same tuples as $a_{r'}$.

In this model, we additionally employ clause-based feature and integrate it with the word context feature. For instance, a relation ("Hawking", "professor-of", "Cambridge") or ("Hawking", "investigator-of", "Cambridge") could be presented by a clause type "Subject-Verb-Complement", while another relation ("Hawking", "born-in", "Cambridge") is in the form of a "Subject-Verb-Adverb" clause. Therefore, considering only entities will fail to predict relations for the tuple ("Hawking", "Cambridge"). We have used clause types in OpenIE (Corro & Gemulla, 2013; Vo & Bagheri, 2018) when extracting surface patterns for the matrix. We can interpolate the confidence for a given tuple and a specific 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 neighborhood model to incorporate clause types, which is presented as follows:

$$\theta^{NC}_{r,t} = \sum_{k} w'_k f'_k\left(a_{r'_{att\oplus clause}}; a_{r_{att\oplus clause}}\right) \tag{8}$$

where w'_k is the weight of the association between $a_{r'}$ and a_r; and f'_k defines a conjunctive feature including clause information between relation a_r and the neighboring relation $a_{r'}$, e.g., $a_{r'_{att\oplus \text{clause}}} = <$ att:"professor"; clause:"SVC" $>$ and $a_{r_{att\oplus \text{clause}}} = <$ att:"researcher"; clause:"SVC" $>$. In this case, the conjunctive feature includes clause and as such each cell of the matri factorization model will include clause information as well.

4.3. Model-based on selectional preference (E model)

Earlier, Riedel et al. (2013) introduced the use of entities in collaborative filtering based on a similar idea to the word context feature but geared specifically for entities. In their method, they employed entities to predict latent relations, hence we refer to it as the *E Model*. The model embeds each entity and relation type into a low dimensional space of size k. For binary relations such as a_r between a pair of entities $e_t = (e_t^1, e_t^2)$, the relation a_r and the arguments e_t^1 and e_t^2, are modeled in a low dimensional space of size k. 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:

$$\theta^{E}_{r,t} = \sum_{k} a_{r,k} e^1_{t,k} + \sum_{k} a_{r,k} e^2_{t,k} \tag{9}$$

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

$$\theta^{EC}_{r,t} = \sum_{k} a_{r,k} e^1_{t,k} v_{t1,r} + \sum_{k} a_{r,k} e^2_{t,k} v_{t2,r} \tag{10}$$

where $v_{t1,r}$ is clause type for argument e^1, and $v_{t2,r}$ is clause type for argument e^2.

4.4. Model-based on the statistical topic models (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 topic. 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 topics, respectively. Therefore, relations can further be learned by their observations within topics. This helps to determine more relations that are missing when learning from directly observed relations. We use Latent Dirichlet

Table 2
Details of the gold standard corpus.

Corpus	Relation types	#relations
Angeli *et al.*'s dataset	40	22,765

Table 3
Effectiveness of T and C on individual models; EC, NC, TC are individual models with clauses; N + T, E + T, F + T are models with T. Bold values indicate best performing model.

Models	Precision (%)	Recall (%)	F-measure (%)
E	48.23	32.41	38.77
EC	51.97	37.02	41.81
E + T	52.20	40.56	45.65
N	44.61	30.18	36.00
NC	48.94	33.11	39.50
N + T	52.47	38.45	44.37
T	46.79	41.70	44.10
TC	54.71	37.02	44.16
F	**58.02**	39.26	46.83
F + T	47.25	**48.45**	**47.82**

Table 4
Experimental results for interpolated models. Bold values indicate best performing model.

Models	Precision (%)	Recall (%)	F-measure (%)
Baseline (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**

Allocation (Blei, Andrew, & Jordan, 2003; Phan et al., 2011) to generate topics, and then embed this information in the matrix. Let $h = \{t_1, t_2, t_3, ..., t_m\}$ be the set of topics in a topic model generated by and and $t_i = \{e_1, e_2, e_3, ..., e_n\}$ a set of entities that appear in each, e.g., the entity "Obama" present in topic t_1 and t_3 and "Hollande" present in topic t_1 and t_5. 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 topic with other cells. This can be more formally represented as:

$$\theta_{r,t}^{T} = \sum_{k} a_{r,k} e_{t,k}^{1} h_{e_1} + \sum_{k} a_{r,k} e_{t,k}^{2} h_{e_2} \tag{11}$$

where h_{e_1} denotes the vector of topics to which argument e^1 belongs to, and h_{e_2} denotes the vector of topics to which argument e^2 belongs to. For instance, for two vectors of topics $h_{e_1} = \{t_1, t_5\}$ and $h_{e_2} = \{t_4, t_6\}$ where $t_1 = \{e_2, e_5\}$, $t_5 = \{e_1, e_3\}$ and $t_4 = \{e_6, e_9\}$, $t_6 = \{e_5, e_7\}$, the representation of $e_{t'}^1 = \{e_4, e_6, e_2, e_5\}$ $e_{t'}^2 = \{e_6, e_9, e_5, e_7\}$ will actually be the a one-hot encoded representation of the $e_{t'}^1$ *and* $e_{t'}^2$ sets, respectively.

Given the fact that using only topics could be noisy for training purposes, we also further augment the topic model with clause-based features. For instance, ("Hollande","France") can be learned from ("Obama","US") if they are present in similar clause types. This could be formulated as:

$$\theta_{r,t}^{TC} = \sum_{k} a_{r,k} e_{t,k}^{1} h_{e_1} v_{h_{e1}} + \sum_{k} a_{r,k} e_{t,k}^{2} h_{e_2} v_{h_{e2}} \tag{12}$$

where $v_{h_{e1}}$ is the clause type for argument e^1, and $v_{h_{e2}}$ is the clause type for argument e^2.

It is important to mention that while models F, N and E have been proposed earlier in (Riedel et al., 2013), our work is focused on proposing and investigating the role of T and C in this process and systematically proposing how these two types of features can be interpolated with other feature types. The objective is to explore whether T and C features are able to address the recall limitation of the state of the art features.

Table 5
Top and additional relation samples. Bold values indicate best performing model.

Top relation samples	org/country_of_headquarter			person/founded			org/city_of_headquarters			person/country_of_birth			org/member_of		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
F + E + N (Baseline)	**79.87**	63.08	70.49	**76.24**	30.61	43.68	76.78	54.76	63.93	**77.42**	81.96	**79.63**	80.19	77.26	78.70
F + EC + NC	75.58	67.75	71.45	75.31	55.01	63.57	**77.24**	56.36	65.17	70.14	**82.12**	75.65	**81.55**	77.83	**79.64**
F + TC + NC	62.06	78.10	69.16	69.96	**76.30**	72.99	72.03	**78.23**	75.00	68.56	78.63	72.25	66.00	78.22	71.59
F + EC + TC + NC	75.55	**79.34**	**77.39**	75.74	75.84	**75.84**	74.98	78.13	**76.52**	74.00	80.02	76.88	63.07	**79.20**	70.22

Additional relation samples	person/parents			org/shareholders			org_political/religious_affiliation			person/spouse			person/school_attended		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
F + E + N (Baseline)	50.25	41.75	45.56	**69.61**	30.35	48.46	**64.87**	42.27	55.47	**63.70**	49.22	55.53	60.50	25.75	36.13
F + EC + NC	**56.54**	54.89	55.70	67.76	35.19	46.32	56.94	57.42	**57.18**	61.19	55.20	58.04	**69.40**	31.46	43.29
F + TC + NC	44.67	**75.78**	56.20	62.72	38.79	47.93	44.46	**77.11**	56.40	53.83	**76.78**	63.28	52.08	**42.33**	**46.70**
F + EC + TC + NC	53.12	62.88	**57.58**	55.38	**48.41**	**51.66**	51.16	69.09	56.73	62.63	76.62	**68.92**	53.77	38.14	44.62

Table 6
Wikipedia and NYTimes datasets[1].

	Relation types	Extracted sentences	Raw documents
Wikipedia	300	5,827	55,000
NYTimes	300	7,388	58,860

Table 7
Characteristics of OpenIE systems.

OpenIEs	Syntactic analysis	Dependency analysis	Clause analysis
ReVerb	+	-	-
OLLIE	+	+	-
ClausIE	-	+	+
LS3RyIE	+	+	+

Table 8
Accuracy of four OpenIE systems tested against PATTY relation patterns.

	Citation	Wikipedia	NYTimes
ReVerb	(Fader et al., 2011)	19.12 %	18.34 %
OLLIE	(Mausam et al., 2012)	29.34 %	27.76 %
ClausIE	(Corro & Gemulla, 2013)	41.67 %	43.01 %
LS3RyIE	(Vo & Bagheri, 2018)	44.46 %	46.44 %

4.5. Model interpolation and parameter estimation

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:

$$\theta_{r,t} = \theta^{F}_{t,r} + \theta^{N}_{t,r} + \theta^{E}_{t,r} + \theta^{T}_{t,r} \tag{13}$$

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 scores 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 $x_{r,t} = \sigma(\theta_{r,t})$.

5. Experiments and evaluation

In order to benchmark our proposed approach, we perform two sets of extensive experiments. In the first experiment, we use the dataset provided by Angeli et al. (2014). The advantage of this dataset is that it already consists of a gold standard of relations that can be used for evaluation purposes. In the second experiment, we use two corpora from Wikipedia and NYTimes. Unlike the dataset from Angeli *et al.*, these two corpora do not have gold standard relation instances. For this reason, we extract silver standard relation instances from these two corpora based on the relation patterns provided by the PATTY project[2]. Furthermore, given our work requires grammatical clause information, we used the work in (Corro & Gemulla, 2013) 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 Corro and Gemulla (2013). 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

[1] https://bitbucket.org/thuanvd/matrixfac-data/downloads
[2] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/patty/

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 topics, we generate and estimate topic models based on LDA through Gibbs Sampling using GibbsLDA++[3]. We optimize three important parameters a, b and number of topics T in the LDA. It is based on the number of topics and the size of the vocabulary in the document collection, which are a = 50/T and b = 0.01, respectively (Phan et al., 2011). 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.

5.1. Relation extraction based on gold standard dataset

In order to benchmark our approach in the first experiment, we employed the dataset[4] proposed in Angeli et al. (2014) described in Table 2. The content of this dataset is comprised of articles from New York Times where each sentence has been annotated with entity tuples and relation types, which are linked to entities from Freebase. This dataset also consists of gold standard relation instances. Note that, we do not use the dataset from Riedel et al. (2013) given the fact that it does not include the original sentences, which prevents us from being able to identify grammatical clauses or learn the topic models as required in our approach.

In our work, we conducted experiments on both individual models and interpolated models for predicting relations as listed in Tables 3 and 4. 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 Yao (2015) for all models that indicate the confidence value to predict a relation. Table 3 shows the detailed performance of each model as well as the combined models in Table 4. As observed in Table 3, 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. Furthermore, the T model allows the identification of both the direct and indirect co-occurrence of relations through the employment of topics and as a result when including the T model, recall and F-measure greatly improve, e.g., the E + T model with F-measure of 45.65% is better than the E model with F-measure of 38.77%; or F model obtained 46.83% in F-measure while F + T obtained 47.82% 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 synergistic contribution of each of these, e.g., F + E + N (being the baseline presented by Riedel et al. (2013)) and F + E + N + T models have an F-measure of 51.9% and 52.21%, respectively.

The results of interpolated models EC + NC, EC + TC, and EC + TC + NC show that each of the models provide advantage in a non-overlapping aspect of the data and hence their interpolation leads to improved performance. 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 note 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%.

It is important to point out why the interpolation of F and C has not been built and included in Table 3. The main reason for this is that the direct inclusion of clause information into the matrix factorization model of the F model leads the model to determine relation similarity solely based on the similarity of their clause type. In other words a model such as F + C would in essence mean that relations that have been seen in the same grammatical clause structures are similar, which is a semantically incorrect assumption. The correct assumption would be, as we have made in the other interpolated models, to consider those relations that have the same word context or selectional preference and also appear in similar grammatical clause structures to be similar. For this reason and for the inaccurate semantic interpretation that F + C would yield, this interpolated model has not been reported in the paper.

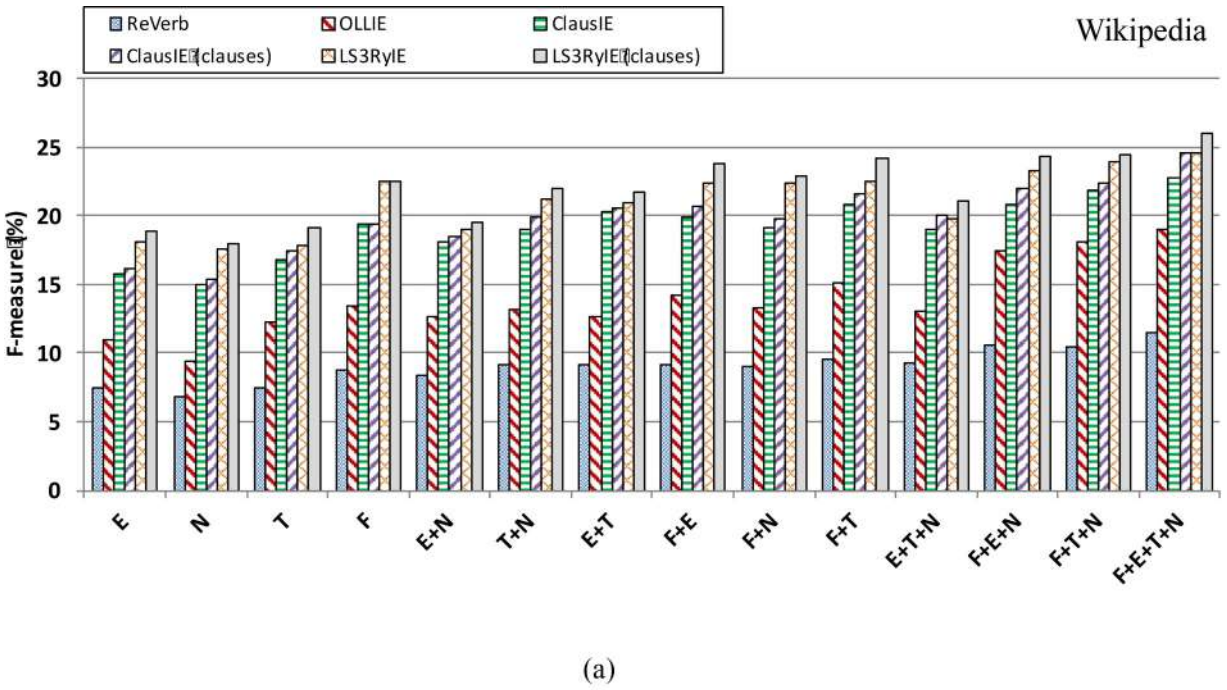
The interpolated models with F such as F + TC, F + NC + TC and F + EC + NC + TC have features, which are derived based on PCA components (F model). Therefore, F + TC, F + NC + TC and F + EC + NC + TC achieve better results compared to the interpolated models without F such as TC, NC + TC, and EC + NC + 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 by Riedel et al. (2013).

Finally, we would like to summarize the impact of our proposed work on performance. As seen in Table 4, when employing clause types on the baseline (F + E + N vs. F + EC + NC), we see that recall increases and overall the incorporation of clause type improves F-measure. Also when adding 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.

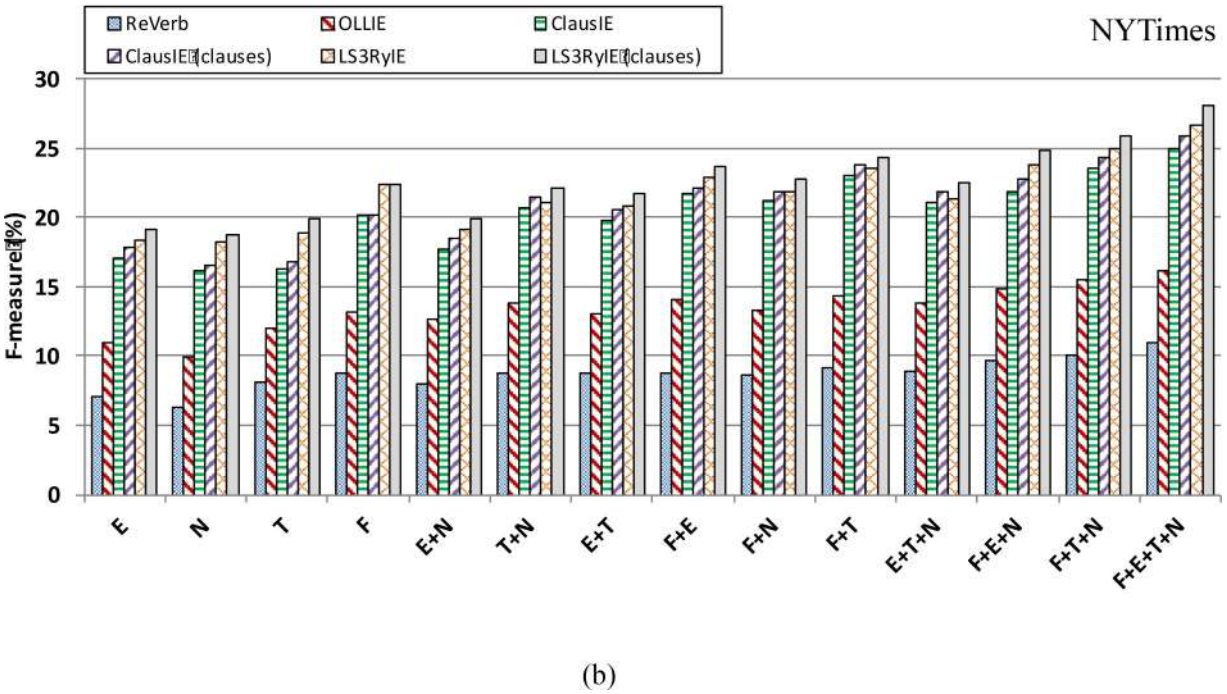
Table 5 shows several relation types, which are broken down into 'top' and 'additional relation sample' types. The top relations are based on pooled results in the matrix models which provide the basis for meta-analysis according to Bravata & Olkin (2001). We report three models where T and C show strong influence compared to the baseline model (F + E + N), namely F + EC + NC, F + TC + NC and F

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

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



(a)



(b)

Fig. 3. The performance of different features based on various OpenIE systems. Wikipedia (a) and NYTimes (b).

+EC+TC+NC. In most of the models, recall and F-measure have greatly improved. While models with E and N have used references for predicting relations on entities, their limitations is in that they are trained for relationships between specific entity tuples and relation types, and therefore, are limited when an insufficient number of explicit evidence is present for each relation type for specific entity tuples. Regarding models with T and C, the relations take advantage of selectional preference in the training process due to their co-occurrence and/or clause type similarity with other relations. These models exploit advanced features from relation characteristics such as clause types and semantic topics for predicting new relation instances. They help in predicting any tuple of entities and relations regardless of whether they were seen at training time with direct or indirect access in their provenance. These results show that the employment of T and C lead to improved recall and hence better performance over F-measure, which has been the objective of our work.

Now, 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. Most of the models have increased 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 their constituting separate models. Comparing our best models (F + NC + TC) and (F + EC + NC + TC) with Riedel *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 Riedel *et al.* achieved 79.58% of precision and 38.51% of recall. Applying topic models 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 Riedel *et al.*'s model.

We would also like to point out that when working with the dataset provided by Angeli et al, we observed an unbalance in the dataset where some relation types occupy a large number of records in dataset while the other relation types are not that prevalent. For instance, "per/country_of_residence" has 2371 instances but "per/country_of_death" has 213 instances. Therefore, when the number of entity pairs are not large enough in the dataset, an impact to the E and N models can be observed. Consequently, the baseline model F + E + N obtained low results on recall and F-measure. However, enriched features in our models such as statistical topic models and grammatical clauses combined with entity and word context features will lead to the identification of an increased number of hidden relations. It should be noted that while our model obtained lower precision than the baseline when exploiting topic and clause features, it showed noticeable improvement on F-measure compared with the baseline. 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/charges" 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 evidence 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 entity tuples occur in the dataset, e.g., the entity tuple < "WASHINGTON", "US" > is seen 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.

5.2. Relation extraction based on silver standard datasets

The second set of experiments is focused on two unlabeled datasets from New York Times and Wikipedia. In order to be able to use these two datasets, we used the relation patterns provided by the PATTY dataset[5] to automatically extract relation instances from these two datasets, which we refer to as the silver standard. As described in Table 6, the two datasets consist of 5,827 sentences of 300 relation types from Wikipedia and 7,388 sentences of 300 relation types from NYTimes. Note that, the NYTimes dataset used here is different from the dataset used in the previous experiment. Now, unlike the first experiment where the gold standard was randomly split into test and train sets, we do not use the silver standard relations extracted using the PATTY patterns in the *training* process. Instead, we use several OpenIE systems to extract relation instances from the two datasets that would then form the training set and will be used for initializing the matrix model. More specifically, we employ four OpenIE systems to extract relation instances for building universal schemas for the matrix model:

- ReVerb (Fader et al., 2011). The system extracts verb phrase-based relations based on a set of syntactic and lexical constraints to identify relations based on verb phrases and then finds a pair of arguments for each identified relation phrase.
- OLLIE (Mausam et al., 2012). The system, an extension of the ReVerb system, uses various heuristics to obtain propositions from dependency parsers. OLLIE performs deep analysis on the identified verb-phrase relations and then extracts all relations mediated by verbs, nouns, and adjectives, among others.
- ClausIE (Corro & Gemulla, 2013). This system exploits 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.
- LS3RyIE (Vo & Bagheri, 2018). The system extends the work by ClausIE with grammatical structure reformulations that help identify discrete relations that are not found in ClausIE, and reduce the number of erroneous relation extractions.

[5] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/patty/

Table 9
F + E + N vs. F + E + N + T on the Wikipedia dataset. Bold values indicate best performing model.

OpenIEs	F + E + N (Baseline)			F + E + N + T		
	P	R	F1	P	R	F1
ReVerb	13.47	8.79	10.62	12.16	10.74	11.46
OLLIE	19.49	15.71	17.40	19.52	18.65	19.01
ClausIE	21.43	20.34	20.87	20.24	25.86	22.71
ClausIE(clauses)	**22.04**	21.79	21.91	**21.81**	29.08	24.03
LS3RyIE	21.77	25.07	23.31	20.73	30.22	24.59
LS3RyIE(clauses)	21.95	**27.25**	**24.32**	21.00	**33.23**	**25.47**

Table 10
F + E + N vs. F + E + N + T on the NYTimes dataset. Bold values indicate best performing model.

OpenIEs	F + E + N (Baseline)			F + E + N + T		
	P	*R*	*F1*	*P*	*R*	*F1*
ReVerb	11.87	8.17	9.71	11.30	10.73	11.02
OLLIE	17.12	13.66	15.20	15.49	16.98	16.21
ClausIE	**23.72**	20.23	21.84	**25.12**	24.85	24.98
ClausIE(clauses)	21.89	23.69	22.75	23.08	29.51	25.90
LS3RyIE	23.59	23.98	23.78	24.39	29.36	26.65
LS3RyIE(clauses)	22.83	**27.32**	**24.88**	24.08	**33.54**	**28.64**

Table 7 presents the characteristics of four OpenIE systems. These systems use different forms of linguistic analysis such as syntactic analysis, dependency analysis and grammatical clause analysis. Please note that ClausIE and LS3RyIE systems are evaluated with both when they considered grammatical clause structures as well as when they did not.

5.2.1. Results

Table 8 depicts the performance of four OpenIE systems on the two datasets when compared to the relation instances that are included in the silver standard produced by the PATTY relation patterns. As expected, the results reveal that clause-based OpenIE systems such as LS3RyIE and ClausIE have a better performance compared to ReVerb and OLLIE. Regarding Wikipedia and NYTimes, LS3RyIE produced 44.46% and 46.44%; ClausIE produced 41.67% and 43.01%; OLLIE produced 29.34% and 27.76%; and ReVerb produced 19.12% and 18.34% in term of accuracy, respectively. Once the relation instances were derived from the raw sentences from the two datasets by the OpenIE systems, we employed the relation instances to initialize the matrix model and perform our proposed matrix completion task, the results of which would then be compared with the relations in the silver standard dataset.

Similar to the first set of experiments, we conducted the evaluations on both individual and interpolated models for predicting relations with surface schemas extracted from the four OpenIE systems. Fig. 3a and 3b show the performance of each model using the four OpenIE systems for Wikipedia and NYTimes datasets. Similar to the discussion in Section 5.1, most of the interpolated models yield better results compared to the individual models. Models E + N, E + T, and E + T + N benefit from additional aspects of the data compared to individual models E, N and T and take advantage of presentation from entities, topic models and related neighbors when referencing the argument slot presentation of the matrix. For the Wikipedia dataset, the best performance was observed using the F + EC + TC + NC model especially for the LS3RyIE system, which incorporates the use of grammatical clauses.

It is easy to see that the models using clause-based OpenIE systems such as LS3RyIE and ClausIE yielded the best performance compared to syntactic-based or dependency-based OpenIE systems (ReVerb or OLLIE). Particularly, the system model F + E + T + N obtained 26.03% and 24.56% of F-measure on LS3RyIE when clause information were considered, denoted as *LS3RyIE(clauses)*[6], and *ClausIE(clauses)*, referring to ClausIE when clause information were taken into account. However, when clause information was not considered, LS3RyIE obtained 24.59% and ClausIE obtained 22.71% in terms of F-measure. Regarding ReVerb and OLLIE, a performance of 19.01% and 11.49% on F-measure was observed, respectively. In the F model and based on LS3RyIE and ClausIE without the consideration of clause information, a performance of 22.84% and 19.46.14 on F-measure was observed, respectively while using OLLIE and ReVerb, we only obtained 12.60% and 8.79% on F-measure. With regards to the NYTimes dataset, similar to Wikipedia, performance of the F + EC + TC + NC model obtained the highest results compared to other models. Particularly, using LS3RyIE (clauses) and ClausIE(clauses), we obtained the highest results with 28.05% and 26.11% while we only obtained a performance of 26.65% and 24.98% with LS3RyIE and ClausIE, respectively, when clause information was not considered.

In comparison, we discuss the performance of the model where T and C showed strong effectiveness compared with the baseline model (F + E + N) proposed by Riedel et al. (2013). Table 9 presents the performance of the two models including F + E + N and F + E + T + N using four OpenIE systems on the Wikipedia dataset. The performance of the F + E + N + T model is better than the baseline

[6] It should be noted that in Figure 3 as well as Tables 9-12, E, N and T are in fact EC, NC and TC when corresponding to the ClausIE (clauses) and LS3RYIE (clauses) rows, because these variations consider clause information. ClausIE and LS3RYIE when mentioned without '(clauses)' do not consider clause information and hence represent all models without interpolation with the C features.

Table 11
Top and additional relation samples on the Wikipedia. Bold values indicate best performing model.

Top relation samples		organization/employer			person/country			government/location			educational_institution/location			person/organization_member		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	50.70	39.00	44.08	53.58	33.78	41.43	51.57	30.07	37.98	45.92	32.29	37.91	54.35	31.75	40.08
	F+T+N	60.02	44.73	51.89	61.34	44.73	51.73	35.81	48.76	41.00	49.83	24.03	32.42	35.81	48.76	41.29
	F+E+T+N	45.03	61.96	52.15	36.29	48.66	41.57	29.41	48.67	36.66	27.23	54.86	36.39	32.45	45.45	37.86
OLLIE	F+E+N	56.16	50.25	53.04	60.40	35.83	44.97	42.02	46.62	44.42	50.73	37.54	43.14	54.28	38.03	44.72
	F+T+N	53.21	54.28	53.73	48.64	58.47	53.10	46.99	47.46	47.22	59.28	37.56	45.98	45.70	57.37	50.87
	F+E+T+N	49.24	73.03	58.82	45.12	74.90	56.31	42.07	54.90	47.63	41.68	52.82	46.59	46.21	67.53	54.87
ClausIE	F+E+N	61.12	63.02	62.05	64.13	60.61	62.32	50.06	57.45	53.50	54.36	45.27	49.40	63.32	57.28	60.14
	F+T+N	62.78	58.23	60.41	60.82	65.96	63.28	48.63	59.40	53.47	43.09	58.48	49.61	60.40	60.78	60.58
	F+E+T+N	60.14	65.13	62.53	64.8	70.00	67.29	50.30	64.24	56.42	54.29	57.38	55.69	60.14	65.13	62.53
ClausIE (clauses)	F+E+N	55.96	58.93	57.41	55.78	64.46	59.81	**70.57**	41.09	51.96	61.11	43.47	50.59	64.46	55.78	59.80
	F+T+N	47.43	71.91	57.16	51.93	74.71	62.76	51.88	66.11	58.12	47.18	57.62	51.55	66.79	67.36	**67.16**
	F+E+T+N	57.35	68.86	62.24	67.55	71.24	**69.35**	55.03	64.14	59.23	51.07	68.49	58.99	56.35	69.53	62.25
LS3RyIE	F+E+N	75.04	54.05	62.83	64.96	54.68	59.37	60.97	57.59	59.23	**68.45**	55.61	61.36	68.71	51.69	58.90
	F+T+N	66.06	82.71	73.43	61.37	71.69	66.12	54.81	78.59	64.59	65.44	80.00	71.99	56.92	65.08	60.72
	F+E+T+N	65.20	**80.90**	72.20	59.24	70.03	64.18	60.14	61.15	60.64	63.05	79.26	70.22	60.00	60.60	60.02
LS3RyIE (clauses)	F+E+N	**77.53**	61.32	68.48	**87.03**	51.99	65.09	55.57	73.89	63.43	70.34	61.69	65.73	**70.06**	57.62	63.23
	F+T+N	76.21	76.93	76.60	63.40	**76.08**	69.16	55.98	**81.67**	**66.43**	68.26	78.60	73.06	64.38	67.47	64.42
	F+E+T+N	76.78	77.78	**77.25**	66.79	67.35	67.06	56.75	80.00	66.39	66.06	**84.00**	**73.95**	55.67	**83.80**	66.20

Additional relation samples		tv_actor/program			project/location			employer/location			governmental_jurisdiction/country			business_operation/citytown		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	35.32	23.96	28.55	30.54	12.73	17.97	29.49	24.84	26.96	29.35	15.99	20.70	39.37	19.65	26.22
	F+T+N	22.26	47.49	30.37	11.66	38.33	17.88	23.18	28.23	25.46	13.50	45.87	20.86	26.49	21.17	23.53
	F+E+T+N	36.30	28.31	31.8	17.77	27.66	21.63	24.33	40.13	30.29	24.76	25.10	24.92	35.15	22.88	27.71
OLLIE	F+E+N	35.79	38.55	37.11	**30.56**	13.72	18.93	40.12	29.27	33.84	32.98	16.07	21.61	40.95	26.97	32.52
	F+T+N	44.22	36.97	40.27	13.48	36.50	19.68	35.58	42.71	38.82	18.85	29.69	23.06	32.06	38.74	35.09
	F+E+T+N	41.19	36.70	38.81	19.70	26.63	22.64	29.01	45.62	35.47	27.14	27.07	27.10	38.47	29.25	33.23
ClausIE	F+E+N	59.49	30.32	40.16	19.70	31.90	24.35	49.14	25.41	33.50	37.80	19.46	25.69	32.32	41.34	36.28
	F+T+N	43.68	42.20	42.92	16.82	46.70	24.73	46.14	32.04	37.81	18.88	43.95	26.41	31.78	47.10	37.95
	F+E+T+N	50.90	42.07	46.07	18.51	**52.89**	27.42	47.29	31.96	38.14	19.63	**49.24**	28.07	31.89	51.08	39.26
ClausIE (clauses)	F+E+N	61.11	43.57	50.87	17.76	38.11	24.22	**50.57**	29.97	37.64	41.66	19.40	26.47	45.66	31.36	37.18
	F+T+N	38.00	52.32	44.02	17.64	46.72	25.61	29.05	62.70	39.70	28.64	24.21	26.24	30.34	54.38	38.95
	F+E+T+N	46.61	49.19	47.62	19.90	38.43	26.22	36.19	50.62	42.20	36.24	30.21	**32.95**	34.87	52.64	41.95
LS3RyIE	F+E+N	**63.82**	38.18	47.77	28.99	21.49	24.68	53.03	26.93	35.72	**37.38**	21.53	27.32	54.00	24.98	34.15
	F+T+N	54.32	51.43	52.84	26.35	28.81	27.15	31.01	50.00	40.41	27.83	44.56	34.26	36.88	53.28	43.58
	F+E+T+N	41.34	59.38	48.74	25.09	28.16	26.53	32.45	52.65	40.15	25.20	44.41	32.15	33.04	**59.61**	42.51
LS3RyIE (clauses)	F+E+N	44.20	50.08	46.95	28.46	25.10	26.74	42.33	38.88	40.53	36.07	23.90	28.75	**60.57**	29.19	39.40
	F+T+N	47.60	**70.91**	56.96	25.19	32.58	**28.41**	33.60	**58.88**	42.78	33.75	25.75	29.21	53.55	34.17	41.71
	F+E+T+N	52.61	62.42	**57.10**	20.10	38.72	26.46	37.23	54.58	**44.26**	28.48	34.24	31.10	52.61	37.90	**44.06**

Table 12
Top and additional relation samples on the NYTimes. Bold values indicate best performing model.

Top relation samples		country/sport_team_location			actor/film			location/employer			country/organization_member			organization/location		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	53.30	59.67	56.30	53.16	32.48	40.32	61.40	31.22	41.39	61.63	29.03	39.46	41.27	41.42	41.34
	F+T+N	45.81	75.78	57.10	27.30	74.56	39.96	35.04	50.00	41.20	30.67	76.07	43.71	29.11	62.54	39.72
	F+E+T+N	47.42	76.92	58.67	30.38	76.10	43.42	27.95	75.40	40.78	34.02	62.38	44.02	40.07	43.84	41.87
OLLIE	F+E+N	50.70	40.39	44.08	53.58	33.78	41.43	54.35	31.75	40.08	51.57	30.07	37.98	45.92	32.29	37.91
	F+T+N	60.02	45.70	51.89	61.34	44.73	51.73	35.81	48.76	41.29	39.63	42.48	41.00	24.03	49.83	32.42
	F+E+T+N	45.03	61.96	52.15	36.29	48.66	41.57	37.64	38.10	37.86	29.41	48.67	36.66	27.23	54.83	36.39
ClausIE	F+E+N	73.76	68.10	70.81	68.60	61.67	64.95	65.61	62.84	64.19	76.33	48.05	58.97	53.13	61.30	56.92
	F+T+N	63.00	77.73	69.59	61.71	69.08	65.18	54.08	74.23	62.57	56.19	75.11	64.28	53.39	68.57	60.03
	F+E+T+N	70.25	**83.56**	76.34	71.21	73.47	72.32	64.11	74.04	69.28	54.33	82.67	65.56	57.87	72.63	64.41
ClausIE (clauses)	F+E+N	75.93	72.09	73.96	76.87	68.98	**72.71**	75.02	66.79	70.66	**79.80**	60.22	68.66	69.13	56.96	62.45
	F+T+N	74.04	74.04	74.57	61.15	74.84	67.30	61.20	69.73	65.18	53.78	82.25	65.02	52.83	**84.11**	64.89
	F+E+T+N	81.24	81.31	81.12	72.37	73.96	73.15	65.03	81.21	72.22	61.94	**82.45**	**70.66**	60.36	65.56	62.85
LS3RyIE	F+E+N	67.61	75.21	71.20	72.27	62.65	67.11	69.95	64.91	67.33	75.19	52.88	62.09	61.81	60.12	60.95
	F+T+N	69.94	69.96	69.95	72.75	62.61	67.30	65.08	69.87	67.39	56.70	75.38	64.71	53.85	69.33	60.61
	F+E+T+N	77.03	80.19	78.56	69.17	72.65	70.86	69.96	75.50	72.62	64.08	76.25	69.63	64.00	67.87	65.87
LS3RyIE (clauses)	F+E+N	72.33	71.61	71.96	**74.96**	59.96	66.62	**82.71**	59.44	69.17	74.08	57.44	64.70	**75.38**	53.62	62.66
	F+T+N	67.75	76.01	71.64	61.59	**79.34**	69.34	63.08	**80.33**	70.66	61.37	76.52	68.11	67.10	68.03	67.56
	F+E+T+N	**83.57**	83.40	**83.48**	66.59	77.17	71.49	72.77	78.77	**75.65**	68.49	71.24	69.83	61.17	79.26	**69.05**

Additional relation samples		award_winner/employer			kingdom/country			person/alternate_name			olympic_participating_country/location			country/statistical_region		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	18.43	18.83	18.37	12.98	33.18	18.66	45.92	32.29	37.92	32.42	25.23	28.37	12.73	30.54	17.96
	F+T+N	14.64	24.95	18.45	11.68	46.27	18.65	24.03	49.83	32.42	25.49	37.82	30.45	11.66	38.33	17.88
	F+E+T+N	17.77	27.66	21.64	14.05	51.65	22.09	27.23	54.86	36.40	31.27	38.67	34.57	13.36	48.38	20.94
OLLIE	F+E+N	22.63	23.96	23.27	24.17	29.00	26.36	**54.63**	42.62	47.88	51.67	50.23	50.93	26.39	28.60	27.45
	F+T+N	16.10	32.42	21.51	17.77	39.83	24.57	41.19	54.98	47.09	51.60	53.74	52.65	20.38	36.23	26.08
	F+E+T+N	22.49	29.83	25.64	27.68	31.18	29.33	46.06	48.52	47.26	48.61	57.46	52.67	30.86	36.08	33.27
ClausIE	F+E+N	16.01	42.92	23.32	28.48	19.35	23.04	38.13	49.26	42.98	**67.85**	42.51	52.27	46.94	19.33	27.38
	F+T+N	15.73	**53.08**	24.26	21.83	27.01	24.15	37.06	56.86	44.87	54.58	56.90	55.71	34.02	25.74	29.31
	F+E+T+N	25.01	37.27	29.93	22.62	39.89	28.86	50.61	52.82	51.69	50.90	70.05	58.96	36.51	28.52	32.02
ClausIE (clauses)	F+E+N	19.83	38.01	26.06	26.51	25.06	25.76	41.37	47.87	44.38	53.30	59.67	56.30	34.95	26.01	29.82
	F+T+N	23.08	41.72	29.71	23.05	38.59	28.86	44.47	54.13	48.82	51.41	60.78	55.70	25.28	58.33	35.27
	F+E+T+N	**29.75**	37.45	33.15	26.39	**39.47**	31.63	42.32	69.03	52.47	48.72	71.21	57.85	35.90	38.78	37.28
LS3RyIE	F+E+N	21.48	50.00	30.05	42.22	20.97	28.02	34.02	62.38	44.03	52.16	50.00	51.05	50.00	35.04	41.20
	F+T+N	21.60	51.01	30.34	35.51	27.26	30.84	34.90	65.96	45.65	50.00	62.76	55.65	28.07	61.50	38.54
	F+E+T+N	29.48	44.04	**35.31**	32.25	30.00	31.08	35.58	**79.05**	48.97	63.06	64.48	**63.76**	30.30	**69.67**	41.94
LS3RyIE (clauses)	F+E+N	27.03	22.12	24.33	**44.52**	19.36	26.98	44.52	53.62	48.65	49.61	57.80	53.40	**51.84**	36.71	43.00
	F+T+N	21.29	50.71	29.98	32.70	25.54	28.68	42.28	61.31	50.47	47.41	70.37	56.65	42.23	52.08	**46.70**
	F+E+T+N	29.20	38.55	33.60	39.55	29.20	**33.60**	51.88	63.59	**57.14**	54.63	**73.89**	62.81	47.06	41.57	44.14

Table 13
The execution time breakdown (in seconds) for different OpenIE systems.

	ReVerb	OLLIE	ClausIE	LS3RyIE
Short sentence	3.380	4.5 + 2.020	0.018	0.060
Medium sentence	3.503	4.5 + 2.472	0.271	0.319
Long sentence	4.094	4.5 + 3.750	1.054	1.856
Mean	3.466	2.747	0.447	0.745

across all OpenIE systems. We obtained 11.16%, 19.01%, 22.71%, 24.03%, 24.59%, 25.47% on F-measure in the F + E + N + T model using ReVerb, OLLIE, ClausIE, ClausIE(clauses), LS3RyIE and LS3RyIE(clauses), respectively. In contrast, the baseline achieved 10.62%, 17.40%, 20.87%, 23.31% on F-measures using ReVerb, OLLIE, ClausIE and LS3RyIE, respectively. The topic model-based feature presented in the T model leads to better results than the baseline when interpolated with other features. Regarding the performance over the NYTimes dataset, shown in Table 10, the baseline succeeded in producing 9.71%, 15.20%, 21.84% and, 23.78% on F-measure by using ReVerb, OLLIE, ClausIE, and LS3RyIE, respectively. However, the F + E + N + T model proposed in this paper produced more accurate results with an improved F-measure of 11.02%, 16.21%, 24.98%, 25.90%, 26.65%, and 28.64% using ReVerb, OLLIE, ClausIE, ClausIE(clauses), LS3RyIE and LS3RyIE(clauses), respectively. Thus, this shows that the proposed model based on the features introduced in this paper provide meaningful improvements over the baseline especially on recall, which translates to an improved overall F-measure. The improvement in recall is one of the main objectives of our work.

Similar to the gold standard dataset, in order to perform more in-depth analysis of our work, Tables 11 and 12 show several specific relation types in three models based on pooled results where T and C show strong influence on the various models including the baseline model in all four OpenIE systems. We report 'top' and 'additional relation sample' types in both Wikipedia and NYTimes corpora where top relations are based on pooled results. These models take advantage of the E, T and N features in the training process due to the co-occurrence information with other relations. Note that, E, T, and N models are considered as EC, TC and NC in ClausIE(clauses) and LS3RyIE(clauses) systems. Now, in terms of the performance of those relation types, in most of the cases, F + E + T + N outperform the F + E + N and F + T + N models as it takes advantage of the presentation of argument slots based on E and T and the co-occurrence information with neighbors based on N. The combination of F, E, T and N outperforms all other models showing the synergistic contribution of each of these features. Relations can benefit from rich co-occurrences with other neighboring relations that can provide strong evidence in the training process for learning hidden relations. Moreover, given their argument slots are presented in a multi-dimensional space, this could increase the number of desirable co-occurrences for referencing other related relations, e.g., F + E + N vs. F + T + N vs. F + E + T + N in ClausIE and LS3RyIE in most of the cases of the relation samples. As such, incorporating more data based on topics within F + E + T + N improves the performance compared to F + E + N on both Wikipedia and NYTimes as it allows semantic topic information to determine relation type similarity. Finally, it should be noted that models that use clause information such as ClausIE(clauses) and LS3RYIE(clauses) show improved performance compared to their counterparts that do not use clause information; hence, pointing to the effectiveness of grammatical clause information for improving recall while maintaining precision.

Table 14
The execution time breakdown (in minutes) for the matrix models.

Models	ReVerb		OLLIE		ClausIE(clauses)		LS3RyIE(clause)	
	Wiki	NYTimes	Wiki	NYTimes	Wiki	NYTimes	Wiki	NYTimes
E	39.4	44.5	41.4	46.8	44.5(47.8)	52.8(59.3)	48.4(53.1)	53.7(62.4)
N	37.2	40.5	35.8	42.4	38.0(41.3)	45.2(49.1)	41.3(45.9)	46.0(51.7)
T	42.5	48.7	43.1	47.7	46.7(49.7)	54.4(60.4)	51.9(55.2)	56.6(63.6)
F	49.1	58.7	55.7	60.3	64.2	69.7	65.6	71.2
Mean	42.1	48.1	44.0	49.3	48.3(50.8)	55.5(59.6)	51.8(55.0)	56.9(62.2)
E + N	85.2	94.7	97.6	113.3	98.6(104.9)	117.2(131.2)	107.2(116.5)	128.6(139.5)
T + N	92.3	105.5	92.0	124.4	100.8(106.1)	132.2(145.3)	112.0(117.9)	135.3(154.2)
E + T	89.5	107.6	90.2	122.2	95.7(104.0)	135.7(142.5)	104.6(115.6)	140.7(151.3)
F + E	95.4	119.2	113.2	131.6	103.7(112.7)	149.8(154.3)	113.9(125.2)	155.3(163.8)
F + N	94.6	116.4	102.5	127.8	110.0(118.2)	146.9(149.6)	124.8(131.4)	145.0(158.8)
F + T	97.5	122.3	105.7	131.9	128.6(135.4)	145.4(156.5)	130.1(135.5)	156.2(164.2)
Mean	92.4	111.0	100.2	125.2	106.2(113.6)	137.8(146.6)	115.4(123.7)	145.0(155.3)
E + T + N	102.4	126.5	103.2	116.9	114.3(119.1)	137.0(145.5)	123.0(127.3)	146.5(150.8)
F + E + N	108.4	128.6	114.5	125.6	128.2(132.1)	141.6(157.8)	133.6(141.8)	154.1(162.5)
F + N + T	113.9	128.9	120.4	148.9	134.2(141.3)	149.8(162.4)	136.7(150.2)	159.3(165.9)
Mean	108.2	128.0	112.7	130.4	125.6(130.8)	135.5(155.2)	131.5(139.8)	153.3(159.7)
F + E + T + N	115.7	131.5	122.1	132.3	129.9(137.2)	153.7(161.8)	136.5(143.7)	162.8(166.4)

5.3. Summary of findings

The objective of our work in this paper has been to explore how relation extraction based on a matrix completion approach can be performed in such a way that high precision relations can be extracted while many relations are retrieved, i.e., a high recall rate is maintained. We introduced features such as those based on statistical topic models as well as grammatical clause structure, which, theoretically-speaking, had the potential to improve recall. Through our extensive experiments, we have made the following observations:

1. The interpolation of grammatical clause structure information with other features improves both recall and precision as shown in Table 3.
2. The interpolation of statistical topic models with other features significantly improves recall rates at the cost of precision as shown in Table 4. However, it should be noted that the overall F-measure metric is improved noticeably.
3. The interpolation of any of the base models with either clause structure and/or statistical topic models consistently improves recall and hence leads to improved F-measure.

As such, we find that the work proposed in this paper is able to address its objective, which was to improve the overall performance of the relation extraction process as well as address the limitation of the earlier work that was related to a low recall. We have shown that our proposed approach improves recall and f-measure while maintaining a reasonable precision.

5.4. Execution time

We have measured the execution time of the different models when we use them to build universal schemas for the matrix model. We first measure the execution time of OpenIE systems that have been used in our experiments. Given the fact that the execution time of such systems can depend on sentence type, we have performed our experiments on 3 different sentence types based on their structure, namely short sentences (simple), medium sentence (borderline complex) and long sentence (complex). We consider short sentences to have 1–2 extracted patterns while medium sentence produce 3–5 patterns. Long sentences, in contrast, are those that have 5 or more patterns. Table 13 shows the detailed execution time. As seen in the table, it takes on average 0.447s to process sentences when clause generation is only used, while it takes 0.745s on average when we include clause generation as well as grammatical structure reformation. In contrast, ReVerb and OLLIE take on average 3.466s and 2.747s, respectively. Note that, OLLIE also requires an additional 4.5s for loading its initial libraries. LS3RyIE undertakes grammatical reformation as a part of its parsing process and as such requires more time compared to ClausIE. It should be noted that our work in this paper works with patterns that are extracted by OpenIE systems at the sentence level where the result of each sentence is independent from other sentences. Therefore, it is possible to easily distribute the processing of the system and immensely scale it as required.

Regarding the execution time performance of the matrix models, our work requires time for executing the matrix factorization process. Table 14 shows the details of the execution time performance of our work for individual models and interpolated models on both Wikipedia and NYTimes corpora. In most of the cases, the execution time performance of each model is based on how many dimensions the matrix model has. Matrices with higher dimensions require more time than lower dimensional matrices. In individual models, the execution time of the N model is the fastest while the F model takes the most amount of time, e.g., the execution of the N model takes 37.2m while the execution of the F model takes 49.1m when performed using ReVerb. For the interpolated models, there is less difference in terms of execution time between different interpolated models as each individual model can be executed in parallel and only interpolated when the results of all individual models are available, e.g., the mean execution time of the interpolated F + E + T + N model takes 153.7m (161.8m) and 162.8m (166.4m) based on ClausIE(clauses) and LS3RyIE(clauses) while the interpolated model F + E requires around 137 m (146.6m) and 145m (155.3m) on the same NYTimes corpus. Based on our experiments, we conclude that it is possible to easily distribute the different models required by our work and hence scale it to large-scale relation extraction scenarios.

6. Concluding remarks and future work

In this paper, we have presented a framework for predicting potential relation instances based on feature enrichments applied to matrix models that are used in a matrix completion process. We have exploited universal schemas that are formed as a collection of patterns from OpenIE systems and relation schemas from pre-existing datasets to build a matrix model in order to use matrix factorization and collaborative filtering to predict relations. While previous systems have trained relations only for entities, we further exploited advanced features such as clause types and statistical topic models for predicting implicit relation instances. Particularly, we exploited clause-based features extracted from OpenIE systems combined with topic models for predicting potentially relevant relation instances. We have carried out extensive experiments on both gold and silver standard datasets. The results of these experiments show that features based on grammatical clause patterns and statistical topic models are able to increase the recall of the relation extraction task while maintaining a reasonable precision, hence leading to an improved overall performance over F-measure when compared to the baseline.

We are interested in extending our work in three main exciting directions in our future work.

- We will explore how neural embedding-based features could be developed that measure relation type and relation argument

relevance and similarity for predicting potentially relevant yet unobserved relation instances. For example, one might be able to learn neural embedding models that determine that a relation type such as “CEO-of” would be more similar to the “Director-of” relation type compared to the “President-of” relation type. Neural embedding models have already shown improved performance on several IR tasks (Chang, 2011; Jansen & Rieh, 2010) and hence could be helpful in improving the performance of the relation extraction task as well.

- We are also interested in defining features based on graph distance and traversal methods such as random walks to establish a measure of relevance between relation types and arguments. In order to achieve this, we will explore how graphs can be formed based on the unification of relation types and/or relation argument entity matching. The additional measure of relation type or argument similarity based on graphs can then augment our matrix completion framework proposed in this paper.
- Finally, while we have developed a silver standard based on the Wikipedia and NYTimes datasets, we are now working on manually labeling a subset of these datasets so that additional gold standards become available for benchmarking in this area that could be used for performing more reliable experimentations.

References

Abacha, A. B., & Zweigenbaum, P. (2016). MEANS: A medical question-answering systems combining NLP techniques and semantic Web technologies. *Information Processing & Management, 51*, 570–594.

Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relation from large plain-text collections. *Proceedings of the fifth ACM conference of Digital Libraries ACM DL 2000* June 02-07, 2000.

Akbik, A., Visengeriyeva, L., Herger, P., Hemsen, H., & Loser, A. (2012). Unsupervised discovery of relations and discriminative extraction patterns. *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)* (pp. 17–32). December 2012.

Angeli, G., Tibshirani, J., Wu, J., & Manning, C. D. (2014). Combining distant and partial supervision for relation extraction. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2012)* October 25-29, 2014.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the Web. *Proceedings of the 20th international joint conference on Artificial Intelligence (IJCAI 2007)* (pp. 2670–2676). 06-12 January 2007.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O (2013). Translating embeddings for modeling multi-relational data. *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)* December 05-10.

Barrio, P., & Gravano, L. (2017). Sampling strategies for information extraction over the deep web. *Information Processing & Management, 53*(2), 309–331 March 2017.

Blei, D., Andrew, Y. Ng., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Bollegala, D., Matsuo, Y., & Ishizuka, Y. (2010). Relational duality: Unsupervised extraction of semantic relations between entities on the web. *Proceedings of the 19nd international conference on World Wide Web (WWW 2010)* April 26-30, 2010.

Bravata, D. M., & Olkin, I. (2001). Simple pooling versus combining in meta-analysis. *Evaluation & the Health Professions, 24*(2), 218–230. https://doi.org/10.1177/01632780122034885.

Bunescu, R., & Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)* October 6-8, 2005.

Chang, E. Y. (2011). *Foundations of Large-Scale Multimedia Information Management and Retrieval.* Springer, Tsinghua University Press.

Collins, M., Dasgupta, S., & Schapire, R. S. (2001). A generalization of principal component analysis to the exponential family. *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic* (pp. 617–624). December 3-8, 2001.

Corro, L. D., & Gemulla, R. (2013). ClausIE: Clause-based open information extraction. *Proceedings of the 22nd international conference on World Wide Web (WWW 2013)* (pp. 355–366). 13-17 May 2013.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D. S., & Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence, 165*(1), 91–134.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., & Mausam (2011). Open information extraction: The second generation. *Proceedings of the 22nd international joint conference on Artificial Intelligence(IJCAI 2012)* (pp. 3–10). 16-22 July 2011.

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. *Proceedings of EMNLP 2011* (pp. 1035–1545). 27-31 July 2011.

Jansen, B. J., & Rieh, S. (2010). The seventeen theoretical constructs of information searching and information retrieval. *Journal of the American Society for Information Sciences and Technology, 61*(8), 1517–1534.

Kambhatla, N. (2004). Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. *Proceedings of the Association for Computational Linguistics (ACL2004)* (pp. 178–181). 21-26 July 2004.

Kang, U., Papalexakis, E., Harpale, A., & Faloutsos, C (2012). Gigatensor, Scaling tensor analysis up by 100 times - algorithms and discoveries. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* August 12 - 16.

Kemp, C., Tenenbaum, J. B., & Griffiths, T. L. (2006). Learning systems of concepts with an infinite relational model. *Proceedings of the 21st national conference on Artificial intelligence 2006 - Volume 1* (pp. 381–388). July 16-20, 2006.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review, 51*, 455–500.

Koren, Y. (2009). Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining 2009* (pp. 426–434). August 24-27, 2008.

Liu, T., Wang, K., Chang, B., & Sui, Z. (2017). A soft-label method for noise-tolerant distantly supervised relation Extraction. *Proceedings of the 2017 conference on Empirical Methods in Natural Language Processing* (EMNLP 2017).

Mausam, Schmitz, M., Bart, R., & Soderland, S. (2012). Open language learning for information extraction. *Proceedings of the 2012 conference on Empirical Methods in Natural Language Processing(EMNLP 2012)* (pp. 523–534). 12-14 July 2012.

Min, B., Grishman, R., Wan, Li., Wang, C., & Gondek, D. (2013). Distant Supervision for Relation Extraction with an Incomplete Knowledge Base. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)* 9-14 June.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association on Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009)* (pp. 1003–1011). 2-7 August 2009.

Nebot, V., & Berlanga, R. (2014). Exploiting semantic annotations for open information extraction: An experience in the biomedical domain. *Knowledge and Information Systems, 38*(2), 365–389 (2014).

Nguyen, D. B., Theobald, M., & Weikum, G. (2017). J-REED: Joint relation extraction and entity disambiguation. *Proceedings of CIKM 2017* November 6-10, 2017.

Oramasa, S., Espinosa-Ankeb, L., Sordoc, M., Saggionb, H., & Serraa, H. (2016). Information extraction for knowledge base construction in the music domain. *Data & Knowledge Engineering, 106*, 70–83. http://dx.doi.org/10.1016/j.datak.2016.06.001.

Pantel, P., & Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and 44th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 113–120). 17-18 July 2006.

Phan, X. H., Nguyen, C. T., Le, D. T., Nguyen, L. M., Horiguchi, S., & Ha, Q. T. (2011). A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering, 23*, 961–976.

Poon, H., & Domingos, P. (2009). Unsupervised semantic parsing. *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP 09)* August

06 - 07.
Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985). A Comprehensive Grammar of the English Language. *Longman*.
Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). Bayesian personalized ranking from implicit feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence 2009* (pp. 452–461). June 19-21, 2009.
Riedel, S., Yao, L., & McCallum, A (2010). Modeling Relations and Their Mentions without Labeled Text. *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2010)* 20-24 September.
Riedel, S., Yao, L., McCallum, A., & Marlin, M. (2013). Relation extraction with matrix factorization and universal schemas. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)* (pp. 74–84). 9-14 June 2013.
Rosenfeld, B., & Feldman, R. (2007). Clustering for unsupervised relation identification. *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM2007)* (pp. 411–418). 6-10 November 2007.
Ryu, P-M., Jang, M-G., & Kim, H-K. (2015). Open domain question answering using Wikipedia-based knowledge model. *Information Processing & Management, 50*, 683–692.
Singhal, A., Simmons, M., & Lu, Z. (2016). Text mining for precision medicine: Automating disease-mutation relationship extraction from biomedical literature. *Journal of the American Medical Informatics Association, 23*(4), 766–772. http://dx.doi.org/10.1093/jamia/ocw041.
Surdeanu, M., Tibshirani, J., Nallapati, R., & Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)* (pp. 455–465). 12-14 July 2012.
Takamatsu, S., Sato, I., & Nakagawa, H. (2011). Probabilistic matrix factorization leveraging contexts for unsupervised relation discovery. *Proceedings of the 15th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2011)* 24-27 May 2011.
Takamatsu, S., Sato, I., & Nakagawa, H. (2012). Reducing Wrong Labels in Distant Supervision for Relation Extraction. *Proceedings of the 50 the Annual Meeting of the Association for Computational Linguistics (ACL 2012)* 8-14 July.
Turney, P. D. (2008). The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research, 33*, 615–655.
Vlachidis, A., & Tudhope, D. (2016). A knowledge-based approach to Information Extraction for semantic interoperability in the archaeology domain. *Journal of the Association for Information Science and Technology, 67*, 1138–1152.
Vo, D. T., & Bagheri, E. (2018). Self-training on refined clause patterns for relation extraction. *Information Processing and Management, 54*(2018), 686–706. https://doi.org/10.1016/j.ipm.2017.02.009.
Vo, D. T., & Bagheri, E. (2017). Open information extraction. *Encycl. Semant. Comput. Robot. Intell. 1*(1), https://doi.org/10.1142/S2425038416300032 1630003 (6 pages).
Xu, F., Uszkoreit, H., & Li, H. (2007). A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity. *Proceedings of the 45th Annual Meeting of the Association for Comput7ational Linguistics (ACL 2007)* (pp. 584–591). June 2007.
Xu, F., Uszkoreit, H., Krause, S., & Hong Li, H. (2010). Boosting relation extraction with limited closed-world knowledge. *Proceedings of the 23rd international conference on Computational Linguistics (COLING 2010)* (pp. 1354–1362). 23-27 August 2010.
Weston, J., Bordes, A., Yakhnenko, O., & Usunier, N. (2013). Connecting language and knowledge bases with embedding models for relation extraction. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1366–1371). 18-21 October.
Wu, F., & Weld, D. S. (2010). Open information extraction using wikipedia. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)* (pp. 118–127). 11-16 July 2010.
Yao, L., Riedel, S., & McCallum, A. (2012). Unsupervised relation discovery with sense disambiguation. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)* (pp. 712–720). 8-14 July 2012.
Yao, L. (2015). *Ph.D Thesis.* University of Massachusetts at Amherst.
Zhang, C., Xu, W., Ma, Z., Gao, S., Li, Q., & Guo, J. (2015). Construction of semantic bootstrapping models for relation extraction. *Knowledge-Based Systems, 83*, 128–1370 July 2015.
Zheng, S., Hao, Y., Lu, D., Bao, H., Xu, J., Hao, H., & Xu, B. (2017). Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing, 257*, 59–66.
Zhou, G., & Zhang, M. (2007). Extracting Relation Information from Text Documents by Exploring Various Types of Knowledge. *Information Processing & Management, 43*, 969–982.
Zhou, G., Qian, L., & Fan, J. (2010). Tree kernel based semantic relation extraction with rich syntactic and semantic information. *Information Sciences, 180*, 1313–1325.
Zouaq, A., Gagnon, M., & Jean-Louis, L. (2017). An assessment of open relation extraction systems for the semantic web. *Information System, 71*, 228–239.