Latent Relational Model for Relation Extraction

Gaetano Rossiello[1](✉), Alfio Gliozzo[2], Nicolas Fauceglia[2], and Giovanni Semeraro[1]

[1] Department of Computer Science, University of Bari, Bari, Italy
gaetano.rossiello@uniba.it
[2] IBM Research AI, Yorktown Heights, NY, USA

Abstract. Analogy is a fundamental component of the way we think and process thought. Solving a word analogy problem, such as *mason* is to *stone* as *carpenter* is to *wood*, requires capabilities in recognizing the implicit relations between the two word pairs. In this paper, we describe the analogy problem from a computational linguistics point of view and explore its use to address relation extraction tasks. We extend a relational model that has been shown to be effective in solving word analogies and adapt it to the relation extraction problem. Our experiments show that this approach outperforms the state-of-the-art methods on a relation extraction dataset, opening up a new research direction in discovering implicit relations in text through analogical reasoning.

Keywords: Information extraction · Distributional semantics

1 Introduction

Relation Extraction (RE) is a very important capability of Natural Language Processing (NLP) systems. It identifies semantic relations between pre-identified entities in text. RE is particularly useful for Knowledge Base Population (KBP), which is the task of populating Knowledge Bases (KBs) whose schemata have been previously defined by a set of types and relations exploiting information from a text corpus, as well as for building KBs from scratch. For instance, if the target relation is `presidentOf`, a RE system should be able to detect an occurrence of this relation between the entities Donald Trump and United States in the sentence *"Trump issued a presidential memorandum for the US"*.

Several methodologies have been proposed to face the RE problem [1,10–12,18,20,21,26,27]. Recently, [6,15,32] propose neural-based models in an end-to-end fashion through increasingly complex architectures.

Although the neural-based RE approaches show good performance, we contend that they present two limitations. First, they do not fit well for limited domains, where only few seed examples are available. Complex architectures have many parameters, therefore they require a considerable amount of training data in order to learn good representations. It is not surprising, because these

P. Hitzler et al. (Eds.): ESWC 2019, LNCS 11503, pp. 283–297, 2019.
https://doi.org/10.1007/978-3-030-21348-0_19

approaches completely rely on the power of deep neural networks that consist of a blind feature learning without considering the linguistic and cognitive insights that this problem requires. Furthermore, the generalization capability of these approaches is limited to the relation types seen during the training phase, thus they are not applicable to discover relations in new domains or in building a new relational data source from scratch.

We approach the RE task from a different angle by addressing it as an analogy problem. Solving analogies, such as ITALY:ROME=FRANCE:PARIS, consists of identifying the implicit relations between two pairs of entities. The research hypothesis that we will be exploring throughout this work is that a method used to recognizing analogies can be useful to discover relations in text. In other words, relation extraction and word analogy are "two sides of the same coin".

These concerns lead to the following research questions: [**RQ1**] *How to address relation extraction as an analogy problem?* [**RQ2**] *Can a relational model be compared with the state-of-the-art RE methods?* In order to answer these questions, we propose an Analogy-based Relation Extraction System (ARES) by exploiting a relational model [28] which still holds the best scores in solving word analogies. Our method projects entity pairs in a relational vector space built by embedding the implicit properties which are observed in the text about how two entities are related.

In this paper, we formalize relation extraction as an analogy problem through its geometric interpretation in the relational vector space. We show that following this idea it is possible to face the RE in different scenarios (unsupervised, semi-supervised, supervised) through the same relational representations. Then, we measure the performance of our approach on a popular dataset designed for distantly supervised RE. The evaluation shows that ARES, with a simple linear classifier, outperforms the previously known approaches. This achievement opens up new promising research directions for relation extraction by exploiting analogical reasoning.

The paper is structured as follows: Section 2.1 describes the state-of-the-art and the recent progress in RE. In Sect. 2.2 we introduce the word analogy problem and the relative approaches. In Sect. 3 we describe ARES and we provide an evaluation of it in contrast with the most popular distant supervised RE approaches in Sect. 4. Section 5 concludes the paper, highlighting the possible new directions for RE.

2 Related Work

2.1 Relation Extraction

Given two entities e_1 and $_2$ that occur in a sentence S, Relation Extraction (RE) is the process to understand the meaning of S and extract a triple $r(e_1, e_2)$, where r represents the semantic relation between the two entities. In the literature several paradigms have been proposed to address the RE problem which differ in terms of input, output and technique adopted, such as pattern-based [10], bootstrapping [1], supervised [12,21,26] or OpenIE [18].

A promising idea, called *distant supervision* [20], consists in using existing KBs, like Freebase [2], as source of supervision without any human intervention. The pairs of entities that belong to a certain relation in the KB are linked with their surface forms in the textual corpus given as input. For each pair, all sentences in the corpus in which the two entities occur together are collected. However, the wrong labeling caused by the automatic matching between the entity pairs in the KB and in the textual content as well as the overlapping relations due to the intrinsic multi-graph structure of the KBs, require more complex training and prediction phases. This paradigm is commonly addressed as a multi-instance [23] and multi-label [11,27] classification task.

The deep neural network models proposed in [6,15,32] attempt to solve the multi-label and/or multi-instance setting in an end-to-end fashion through neural-based architectures with the aim to avoid the error propagation that could be raised by the use of lexical and syntactic tools for feature extraction.

Another method, so-called universal schema [24,30], faces RE by combining the OpenIE and KB relations. This method is related to our, in the sense that a pair-relation matrix is built, but it differs from the idea. Indeed, the goal of the universal schema is to address RE using a collaborative filtering approach typically adopted in recommender systems.

2.2 Word Analogy

The word analogy task, namely the proportional analogy between two word pairs such as $a : b = c : d$, has been popularized by [19] with the aim to show the capability of their neural-based model, so-called word2vec, in discovering the "linguistic regularities" just using vector offsets ($king - man + woman = queen$ is the most cited example). Several studies [14,16] have been proposed to deeply analyze the use of word embeddings and vector operations in attempting to achieve better performance on the same Google analogy dataset. The works in [5,31] explore the use of word vectors to model the semantic relations.

However, the word analogy task has been originally addressed by [29] who investigate several similarity measures on Scholastic Aptitude Test (SAT) dataset, composed of 374 multiple-choice analogy questions. Given $mason : stone$, this task consists of selecting the right analogy among 5 possible choices ($carpenter : wood$ in this case). The authors provide an interesting argumentation regarding the different types of similarities, *attributional* and *relational*, and their use in facing the word analogy problem. The lesson learned is that the attributional similarity, typical of the word space models [13,22,25], is useful for synonyms detection, word sense disambiguation and so on. Instead, the relational similarity fits better in understanding word analogies. This intuition is confirmed by [3] who shows that word2vec is less effective on the SAT dataset. Conversely, the relational model proposed in [28] achieves a performance (56.1%) close to the human level (57.0%) on the same benchmark. Therefore, in this work we extend and adapt this relational model in order to address the relation extraction problem.

3 Methodology

In this section we present ARES and we explore its use to face the RE problem through analogical reasoning. First, we describe the Latent Relational Model (LRM), the foundation of our method. Then, we show that an extensional representation of the relations can be provided through the geometric interpretation of analogy between entity pairs. Finally, we explore the application of ARES to different RE scenarios.

3.1 Latent Relational Model

LRM provides an intensional representation of relations by embedding the implicit properties observed in the text about how two entities are related. This idea relies on the *distributional hypothesis* [9] which finds its roots in psychology, linguistics and statistical semantics: *"linguistic items with similar distributions have similar meanings"*.

Given a textual corpus T, the aim is to build a vocabulary V, composed of the unique entity pairs extracted from T, and a lookup table $M^{n,k}$, with $n = |V|$, consisting of k-dimensional latent relational vectors associated to each element of V. The idea to build a relational vector space model was originally proposed in [28,29] to solve a word analogy task. We extend and adapt it to address the RE problem. The main differences concern the use of an entity-entity vocabulary, instead of a word-word one, and a different way to extract the contexts around a pair as explained in the following paragraphs.

Entity Pair Vocabulary. Given a textual corpus T, the first step is to build a vocabulary $V = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, where (X_i, Y_i) are the distinct entity pairs that occur together at least in one sentence. The question is how to identify the atomic lexical units in T that are considered as entities (X_i, Y_i). This can be done in different ways based on the specific RE scenario. For instance, in an unsupervised RE a Named Entity Recognizer (NER) or, more generally, a noun phrase chunker can be adopted. It depends from the types of relations to be extracted. In a distant supervised RE, V can be built using entities coming from the KB linked in the text.

Entity Pair Contexts. Once the vocabulary V is built, the next step is to extract the contexts around each entity pair when they occur together into the same sentences across the corpus T. A careful choice of the contexts is fundamental because they are the properties that define the intensional representation of a relation. Differently from [28,29], we adopt a richer set of lexical and syntactical features extracted from each sentence as proposed in [20].

Given an entity pair, from each sentence in which the pair occurs we extract:

1. The entity types provided by the NER;
2. The sequence of words between the two entities;

3. The part-of-speech tags of these words;
4. A flag indicating which entity came first;
5. An n-gram to the left of the first entity;
6. An n-gram to the right of the second entity;
7. A dependency path between the two entities.

If an entity pair occur in more than one sentence, we collect the features extracted from each sentence into a single bag. It should be noted that this may involve the wrong labeling issue using a distant supervised approach, which requires a multi-instance setting to be addressed [23]. Instead, in our model the context aggregation helps to provide a more accurate intensional representation of the relations between an entity pair.

Relational Vector Space Model. In this step a sparse matrix $X^{n,m}$ is built by mapping the n entity pairs in V to the rows and the m distinct features/contexts extracted in the previous step to the columns. Each element $X_{i,j}$ represents the weight of the j-th context in relation to the i-th entity pair. This weight might be computed using different well-known weighing schemes in information retrieval [4] and distributional semantic models [13], such as binary, tf-idf, entropy and so on. Indeed, our *pair-context* matrix is the relational version of the classic *document-term* or *term-term* vector space models.

There is not a theoretical motivation about which weighing schema is better: the choice is empirical, and depends on the specific purpose and on the distribution of the information in the textual corpus. In our experiments we found that when applied to the RE task, tf-idf weights tend to produce more precise results, while the binary schema achieves a recall-oriented performance.

Matrix Factorization. Since $X^{n,m}$ is a highly sparse matrix, this representation is not able to catch the implicit meaning across the textual contexts which express the same semantics. For instance, the phrases *"A is the author of B"* and *"C wrote D"* have the same meaning w.r.t. the relation `authorOf`, but the patterns *"is the author of"* and *"wrote"* are represented as separate features in X. As consequence, the vectors related to the pairs (A,B) and (C,D) in X are orthogonal even if they convey the same concept. In line with [4,13,28], we address this issue by applying Singular Value Decomposition (SVD) to the sparse matrix X.

SVD decomposes a matrix X into a product of three matrices $U\Sigma V^T$, where $U^TU = I = V^TV$ and Σ is a diagonal matrix of sorted singular values having the same rank r of X. Let Σ_k, with $k \ll r$, be the truncated version of Σ by considering only the first k singular values, the SVD finds the best matrix $X_k = U_k\Sigma_kV_k^T$ by minimizing the cost function $||X - X_k||_F$. We adopt the fast and scalable algorithm proposed in [8].

Thus, the SVD applied to X produces a low-rank approximation of X:

$$X \approx X_k = U_k\Sigma_kV_k^T \tag{1}$$

where k is a hyper-parameter. For our purpose, we are mainly interested in the matrices $(U_k \Sigma_k)^{n,k}$ and $V^{k,m}$. Indeed, the lookup table $M^{n,k}$ that we are looking for is obtained by:

$$M^{n,k} = (U_k \Sigma_k)^{n,k} \tag{2}$$

Each i-th row in M is a k-dimensional latent relational vector associated to each entity pair in V. SVD allows to take into account the global distribution of the pair contexts in the corpus and to *understand* the implicit relationships among them. This latent information is embedded into the k-dimensional dense vectors.

On the other hand, $V^{k,m}$ contains the latent vectors of each m feature/context. Thus, the SVD has the big advantage of projecting the pairs and the contexts into the same vector space. The role of $V^{k,m}$ is crucial for two reasons. Firstly, in a supervised RE the k-dimensional vectors of new entity pairs in the test set are obtained by $M^{n,k} = XV_k^T$ without retraining the SVD. However, the most interesting aspect regards *transfer learning* domain adaptation: the SVD can be applied to a pair-context matrix X^{Web} build on a large web scale corpus, so V_k^{Web} condenses a rich prior knowledge that can be infused into a new domain just using a matrix multiplication [7].

Finally, many other techniques can be applied to solve the sparsity issue, such as Non-negative Matrix Factorization (NMF) or deep neural network, like Auto-Encoders (AE) that learn latent representation through a non-linear dimensionality reduction. A comprehensive comparison of all these methods as well as the application of transfer learning for domain adaptation are out of the scope of this work, but, surely, they represent a very promising directions for future investigations.

3.2 Geometric Interpretation of Analogy

Through LRM, each entity pair occurring in the corpus is projected into a relational vector space, therefore it is possible to exploit its geometric interpretation to measure similarities between entity pairs. Thus, we can assert that there is an *analogy* between two pairs of entities (A, B) and (C, D) iff their latent vectors are close in the relational vector space. For instance, we can measure this proximity with the angle between the relational vectors using the cosine similarity.

Formally, given $r_{(A,B)}$ and $r_{(C,D)}$ the relational vectors in M related to the entity pairs (A, B) and (C, D):

$$A : B = C : D \Leftrightarrow cosine(r_{(A,B)}, r_{(C,D)}) > t \tag{3}$$

where

$$cosine(r_{(A,B)}, r_{(C,D)}) = \frac{r_{(A,B)} \bullet r_{(C,D)}}{||r_{(A,B)}|| \cdot ||r_{(C,D)}||} \tag{4}$$

and t is a threshold that establishes the breadth of the analogy between the two pairs.

This intensional representation of the relations well models the fuzzy meaning of *relation* between two entities. In fact, let us first consider the boundary cases with the cosine similarity equal to 1 and 0. If 1 it means that (A, B) and (C, D)

share exactly the same properties observed in the text, therefore the pairs are strictly analogous. Instead, the value 0 means that their vectors are orthogonal, so we can state that the pairs are not analogous at all[1]. However, since the range of the cosine is $[-1, 1]$, infinite degrees of analogy might be defined between two entity pairs, and this aspect depends on the value of the threshold t in Eq. (3).

This is useful to define the granularity of the type of a relation: higher values of t mean fine-grained relations, otherwise lower values mean relations that are more inclusive and coarse-grained. For instance, given the following sentences: (1) Rome *is the capital of* Italy; (2) *The capital of* France *is* Paris; (3) Brooklyn *is a borough of* New York. Into a hypothetical vector space, the latent vectors $r_{(Italy,Rome)}$ and $r_{(France,Paris)}$ are close because they share the same context *"capital of"*. On the other hand, $r_{(NewYork,Brooklyn)}$ is farther to the other two vectors, but it is not orthogonal because the concept of *"borough of"* is semantically related, in some way, to *"capital of"*. Indeed, both patterns *"borough of"* and *"capital of"* express the meaning of inclusion between two locations. Therefore, we can say that Italy:Rome=France:Paris. But what about Italy:Rome=New York:Brooklyn? This depends on the granularity of the relation that we are taking into account. If we want to model the relation `capital`, then we can say that (Italy, Rome) and (New York, Brooklyn) are not analogous. Instead, the result changes if we imagine a coarse-grained relation like `contains`. The different scopes of `capital` and `contains` depend on the value of the threshold t.

3.3 Relation Extraction as Analogy Problem

Our aim is to use the geometric interpretation of analogy in attempting to emulate the task in identifying tuples in texts that share the same relations. Formally, given as input a textual corpus T and a semantic relation R, the problem of RE is to extract all pairs of entities that have the relation R in the corpus. Therefore, the output of RE is an extensional representation of the relation R by listing all entity pairs in the corpus that belong to R. The question is: how is R defined in T? Let us consider M_T as the LRM built on the corpus T. Based on the geometric interpretation of analogy described in Sect. 3.2, we can define the relation R in an extensional way through the intensional vector representations in M_T as follow:

Definition 1. *A semantic relation R is a region in a relational vector space M_T that outlines the boundaries among those entity-pair vectors that are analogous to each other.*

Since computing the analogy, hence the similarity, of all possible combination of the entity pair vectors is infeasible, RE is reduced to an optimization problem in finding the boundaries of that *region* in M_T. In the next paragraphs we show the use of ARES to address the different RE scenarios.

[1] Based on the world described in the textual corpus.

Unsupervised Relation Extraction. In absence of training examples, a clustering algorithm can be applied on M_T in order to find $C_1 \dots C_n$ centroids. For instance, the k-means or DBSCAN algorithms can be used depending on if we want to fix or not the number of the centroids. A centroid C_i represents the relational vector that condenses the *meaning* of a relation R_i. Thus, given the relational vectors of the entity pairs in M_T, the relation R_i in the corpus T is defined as follow:

$$R_i = \{(A, B) \mid cosine(r_{(A,B)}, C_i) > t\} \tag{5}$$

The value of t is user-defined parameter that determines the scope of the region around the centroid vector and so the granularity of the relation R_i.

Semi-supervised Relation Extraction. ARES can be adopted also when a small set of few seed pairs that express a relation R is provided as input. Let $R_I = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ a set of seed pairs with n small, then the centroid vector is obtained by averaging the relational vectors in M_T related to each input pair as follow:

$$C_{R_I} = \frac{1}{n} \sum_{i=1}^{n} r_{(X_i,Y_i)} \tag{6}$$

In this few-shot setting, ARES can be applied in an information retrieval style by finding the nearest neighbors of the centroid C_{R_I} used as a query. Thus, the pairs of entities in the corpus T that have the relation R are extracted as follow:

$$R_O = \{(A, B) \mid cosine(r_{(A,B)}, C_{R_I}) > t\} \tag{7}$$

The entity pairs are ranked based on the similarity with the centroid/query C_{R_I} and a user can fix the value of t in order to cut the pairs that have a similarity below that threshold.

Supervised Relation Extraction. In a supervised RE setting a bigger training set of seed entity pairs is available. In particular, the distant supervision ensures a large amount of training data by exploiting existing relational data sources, like Freebase, without any human intervention. Since an entity pair can belong to more relations at the same time, the distant supervised RE is commonly addressed as a multi-label classification task where each relation is a class.

In this setting, ARES exploits the training set in order to find that *region* where the entity-pair vectors are analogous to each other, as stated in Definition (1). For instance, a Support Vector Machine (SVM) classifier trained on the relational vectors of the entity pairs in the training set finds a hyperplane into the hyperspace defined by M_T. In fact, the hyperplane splits the region into the vector space M_T by grouping the analogous entity pair vectors for a specific relation. During the test phase, a new entity pair is projected into M_T and the classifier predict at which region the new instance belong.

4 Experiment

This section describes our evaluation by providing a comparison with the state-of-the-art methods and a further analysis in order to show the flexibility of our approach.

4.1 Experimental Setting

We evaluate ARES on the real-world dataset NYT10 [23], that is commonly used by the community to evaluate the distant supervised RE methods. This dataset was created by aligning Freebase tuples with the New York Times (NYT) corpus from the years 2005–2007.

We adopt the original held-out setting that consists of 51 relations/classes. The training set has 4700 positive and 63596 negative relation instances. While the test set has 1950 positive and 94917 negative examples. We build our LRM using only the sentences in the training set. For LRM we adopt the binary weights and we fix to 2000 the dimension of the relational vectors. The pair contexts are extracted as explained in Sect. 3.1.

We train a set of linear SVM on LRM in a *one-vs-rest* multi-label setting with a penalty equal to 10, chosen with a 3-fold stratified cross validation on the training set. In the prediction phase, we first project the unseen entity pairs into the latent space using LRM as described in Sect. 3.1, then we predict the scores for each relations/classes based on the decision functions of the SVMs.

We evaluate the performance using Precision-Recall curve and P@n metrics. However, during our experiments we tried also other non-linear functions, such as polynomial and rbf kernels, and a Multi-Layer Perceptron (MLP) with a sigmoid function as last layer to avoid the *one-vs-rest* strategy. These classifiers show more stable performance across the classes compared with a linear SVM when learned on our LRM. However, using a simple classifier, without many (hyper)parameters, allow us to evaluate more easily the quality of our relational representations, that is the *research question* of this study.

4.2 Results and Discussion

We compare ARES with the popular feature-based and neural-based distant supervised RE approaches. **MINTZ++** [20] is the first distant supervised method for open domain KB that uses a logistic regression classifier. We adopt the multi-label version. **MIML-RE** [27] is a multi-instance multi-label approach using a probabilistic graphical model to address the wrong labeling issue. **PCNN+ONE** [32] uses a convolutional neural network for sentence representations with the *at-least-one* strategy for multi-instance. **PCNN+ATT** [15] improves the previous deep architecture by adding a sentence-level attention to face the multi-instance learning.

Figure 1 shows the precision-recall curves of each model. The curves proof clearly that our approach outperforms consistently all the state-of-the-art methods with a particular emphasis on the boosted precision at the first part of the

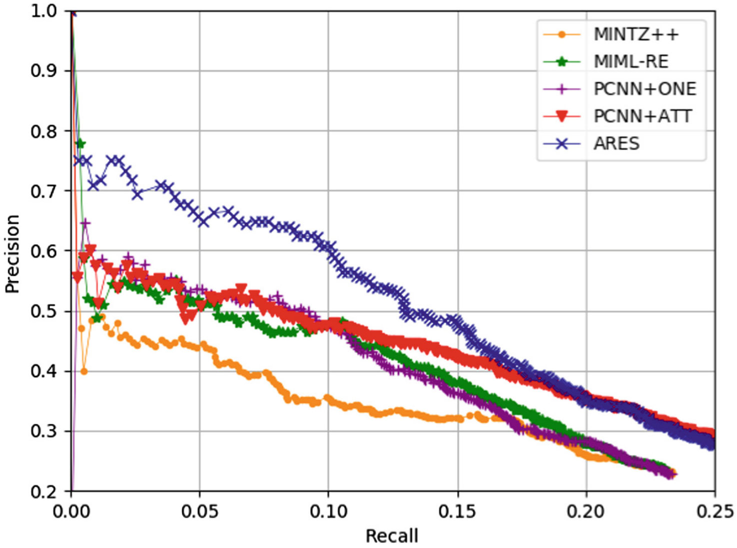

Fig. 1. Precision-Recall curves comparison on NYT10 dataset.

Table 1. Precision values for the top extracted entity pairs.

	P@10	P@100	P@1000	AvgPr
MINTZ++	0.55	0.46	0.32	0.08
MIML-RE	0.64	0.55	0.34	0.10
PCNN+ONE	0.64	0.55	0.33	0.10
PCNN+ATT	0.64	0.55	**0.37**	0.12
ARES (only syntactical)	0.38	0.55	0.29	0.10
ARES (only lexical)	**0.82**	0.66	0.33	0.13
ARES	0.70	**0.68**	0.36	**0.14**

curve. Table 1 shows this aspect with more detail. In fact, ARES achieves a P@100 equal to 0.68, while the other multi-instance methods obtain 0.55.

However, this improvement remains constant along the curve as showed by the average precision in Table 1. ARES achieves these performances just using a simple linear classifier against more complex deep learning architectures. Therefore, our latent relational vectors promote the generalization capability of a classifier. We performed an ablation test over the lexical and syntactical features groups and their combination. As showed in Table 1, the SVM classifier trained only on the lexical group has an average precision very close to that obtained by training the classifier on the full set of features. This result suggests that our approach can be applied also on web-scale corpora since the extraction of the lexical features can be done efficiently.

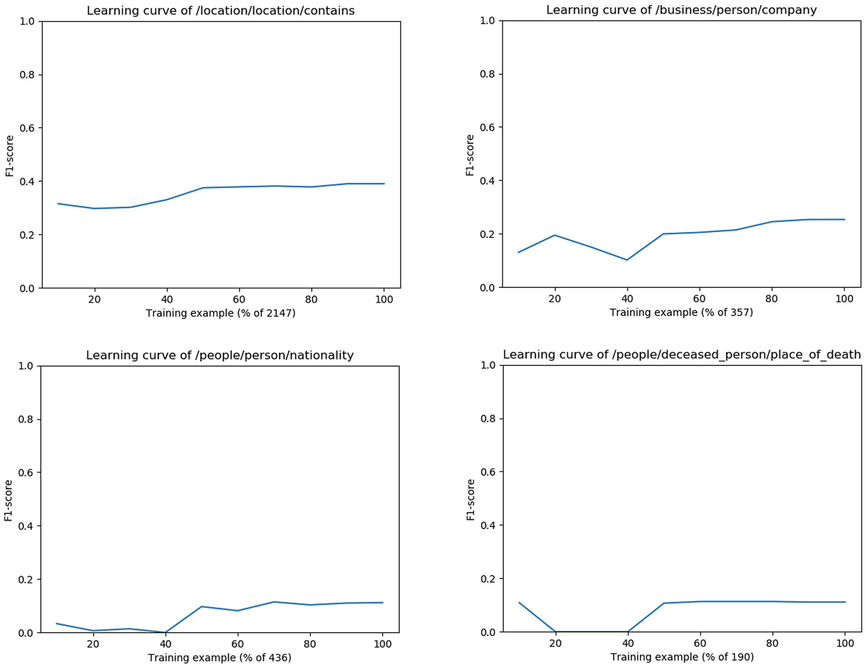

Fig. 2. Learning curves by training the SVMs on different size on the training set.

Moreover, this dataset is highly unbalanced, therefore an end-to-end model trained on this setting tends to overfit on the most frequent relations, like `contains`, and provides a poor representation for the others. Our model alleviates the overfitting because LRM learns the entity pairs vectors in a unsupervised way by taking in account the global distribution of the contexts across the entire corpus. That means better representations also for those relations with few examples, therefore better generalization capability for the classifier.

To confirm this aspect, Fig. 2 shows the learning curves obtained by training the SVMs on different size of training set. We performed this analysis on four frequent relations of the NTY10 dataset by randomly choosing the different buckets of the training instances for each relation. For relations, such as `contains` and `company`, our model reaches almost the best F1-scores just using about the 20% of training examples.

However, it is worth to note that the attention mechanism of PCNN+ATT shows a robust behavior when the recall increases. This suggests that a combination of our LRM with deep neural networks represents an interesting direction for future investigations.

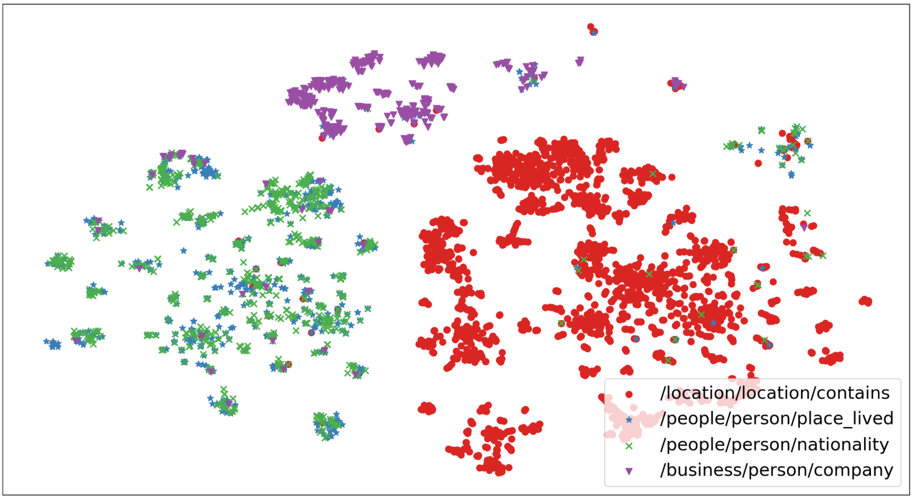

Fig. 3. 2D visualization, using t-SNE [17], of entity pair embeddings learned on the textual corpus of the NYT10 [23] dataset. Each point represents an entity pair vector learned from text through LRM. The entity pairs are aligned with the relation types in Freebase. Each marker represents a different relation type. The distribution of the entity pair embeddings in the vector space approximates the relational structure of the knowledge graph. (Color figure online)

4.3 Unsupervised Relational Analysis

Since LRM is an unsupervised model we can exploit the relational vectors to understand the distribution of the relations in a given textual corpus. Figure 3 shows the 2D projection of the relational representations using t-SNE [17] a techniques used to visualize high-dimensional embeddings. We built a LRM on the whole NYT10 corpus (train+test) and each point in the space is a entity pair vector. For instance, a (red) point marker in Fig. 3 refers to an instance of the relation `location/location/contains`, such as (New York, Brooklyn).

Since the entity pairs are aligned with those in Freebase, we can label them with their relations used as ground truth. As we can see from the figure, the distribution of the entity pair clusters is very close to the ground truth. For instance, the cluster consisting of the (purple) triangle markers represents a group of entity pair vectors with well-defined boundaries and with a strong overlap with the instances of the relation `business/person/company`. Similar behavior occurs for the (red) point markers and the instances of the relation `location/location/contains`. This shows that the LRM is able to produce latent vectors for each entity pair, learned from a corpus, which approximate the relational structure of a knowledge graph like Freebase.

However, there is a strong overlap for certain relations, such as `people/person/nationality` and `people/person/place_lived`. In fact, they are strongly related, but this does not necessary mean that LRM provides poor representations. Instead, we can conclude that the properties in the text are not enough to discriminate the semantics of these relations, hence those in overlap can be removed or merged. In summary, this study shows that LRM is a flexible tool, e.g., also to analyze a corpus and to establish if it is proper or not in application to distant supervision paradigm.

5 Conclusion and Future Work

In this work we explored the use of analogical reasoning to address the problem of extracting relations from textual corpora. We extended a model proposed to solve word analogies in order to provide relational representations that have been proven to be effective for a relation extraction system. Indeed, our approach, using a simple linear classifier, achieves promising results when compared with state-of-the-art deep neural-based models. In our research agenda, we plan to learn non-linear relational representations from text using unsupervised deep neural networks, such as auto-encoders, as well as to explore the use of analogy in transfer learning in order to address more challenging problems, such as domain adaption and automatic ontology construction.

Acknowledgement. This work was conducted during an internship at the IBM Thomas J. Watson Research Center in Yorktown Heights, NY, USA. We thank Anastas Stoyanovsky, Steven Pritko and Gabe Hart, software engineers at the IBM Watson Groups in Pittsburgh and Denver, USA, for helping and inspiring us during the “Fast Domain Adaptation in IBM Watson Discovery” project.

References

1. Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: ACM DL, pp. 85–94 (2000)
2. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD Conference, pp. 1247–1250. ACM (2008)
3. Church, K.W.: Word2vec. Nat. Lang. Eng. **23**(1), 155–162 (2017)
4. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS **41**(6), 391–407 (1990)
5. Gladkova, A., Drozd, A., Matsuoka, S.: Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn’t. In: SRW@HLT-NAACL, pp. 8–15. The Association for Computational Linguistics (2016)
6. Glass, M., Gliozzo, A., Hassanzadeh, O., Mihindukulasooriya, N., Rossiello, G.: Inducing implicit relations from text using distantly supervised deep nets. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11136, pp. 38–55. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_3

7. Gliozzo, A.M., Strapparava, C.: Semantic Domains in Computational Linguistics. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-68158-8
8. Halko, N., Martinsson, P., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review **53**(2), 217–288 (2011)
9. Harris, Z.: Distributional structure. Word **10**(23), 146–162 (1954)
10. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: COLING, pp. 539–545 (1992)
11. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L.S., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: ACL, pp. 541–550. The Association for Computer Linguistics (2011)
12. Jiang, J., Zhai, C.: A systematic exploration of the feature space for relation extraction. In: HLT-NAACL, pp. 113–120. The Association for Computational Linguistics (2007)
13. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes **25**(2–3), 259–284 (1998)
14. Levy, O., Goldberg, Y.: Linguistic regularities in sparse and explicit word representations. In: CoNLL, pp. 171–180. ACL (2014)
15. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: ACL. The Association for Computer Linguistics (2016)
16. Linzen, T.: Issues in evaluating semantic spaces using word analogies. In: RepEval@ACL, pp. 13–18. Association for Computational Linguistics (2016)
17. Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
18. Mausam, Schmitz, M., Soderland, S., Bart, R., Etzioni, O.: Open language learning for information extraction. In: EMNLP-CoNLL, pp. 523–534. ACL (2012)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)
20. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL/IJCNLP, pp. 1003–1011. The Association for Computer Linguistics (2009)
21. Nguyen, T.H., Grishman, R.: Relation extraction: perspective from convolutional neural networks. In: VS@HLT-NAACL, pp. 39–48. The Association for Computational Linguistics (2015)
22. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543. ACL (2014)
23. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6323, pp. 148–163. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15939-8_10
24. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: HLT-NAACL, pp. 74–84. The Association for Computational Linguistics (2013)
25. Sahlgren, M.: An introduction to random indexing (2005)
26. Sun, L., Han, X.: A feature-enriched tree kernel for relation extraction. In: ACL, vol. 2, pp. 61–67. The Association for Computer Linguistics (2014)
27. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: EMNLP-CoNLL, pp. 455–465. ACL (2012)

28. Turney, P.D.: Similarity of semantic relations. Comput. Linguist. **32**(3), 379–416 (2006)
29. Turney, P.D., Littman, M.L.: Corpus-based learning of analogies and semantic relations. Mach. Learn. **60**(1–3), 251–278 (2005)
30. Verga, P., McCallum, A.: Row-less universal schema. In: AKBC@NAACL-HLT, pp. 63–68. The Association for Computer Linguistics (2016)
31. Vylomova, E., Rimell, L., Cohn, T., Baldwin, T.: Take and took, gaggle and goose, book and read: evaluating the utility of vector differences for lexical relation learning. In: ACL. The Association for Computer Linguistics (2016)
32. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: EMNLP, pp. 1753–1762. The Association for Computational Linguistics (2015)