Learning Relational Representations by Analogy using Hierarchical Siamese Networks

Gaetano Rossiello[†], **Alfio Gliozzo**[‡], **Robert Farrell**[‡], **Nicolas Fauceglia**[‡], **Michael Glass**[‡]
[†]Department of Computer Science, University of Bari, Italy
[‡]IBM Research AI, Yorktown Heights, NY, US
gaetano.rossiello@uniba.it, gliozzo@us.ibm.com,
robfarr@us.ibm.com, nicolas.fauceglia@ibm.com, mrglass@us.ibm.com

Abstract

We address relation extraction as an analogy problem by proposing a novel approach to learn representations of relations expressed by their textual mentions. In our assumption, if two pairs of entities belong to the same relation, then those two pairs are analogous. Following this idea, we collect a large set of analogous pairs by matching triples in knowledge bases with web-scale corpora through distant supervision. We leverage this dataset to train a hierarchical siamese network in order to learn entity-entity embeddings which encode relational information through the different linguistic paraphrasing expressing the same relation. We evaluate our model in a one-shot learning task by showing a promising generalization capability in order to classify unseen relation types, which makes this approach suitable to perform automatic knowledge base population with minimal supervision. Moreover, the model can be used to generate pre-trained embeddings which provide a valuable signal when integrated into an existing neural-based model by outperforming the state-of-the-art methods on a downstream relation extraction task.

1 Introduction

The task of identifying semantic relationships between entities in unstructured textual corpora, namely Relation Extraction (RE), is often a prerequisite for many other natural language understanding tasks, e.g. automatic knowledge base population, question answering, etc. RE is commonly addressed as a classification task (Bunescu et al., 2005), where a model is trained to classify relation mentions in text among a predefined set of relation types. For instance, given the sentence "Robert Plant is the singer of the band Led Zeppelin", an effective RE system might extract the triple `memberOf`(ROBERT PLANT, LED ZEPPELIN), where `memberOf` is a relation label expressed by the linguistic context *"is the singer of the band"*.

Since a given relation can be expressed using different textual patterns surrounding entities, the state-of-the-art RE models which follow this approach need a considerable amount of examples for each relation to reach satisfactory performance. Distant supervision (Mintz et al., 2009) instead uses training examples from a knowledge base, guaranteeing a large amount of (popular) relation examples without human intervention, which can be used effectively by neural networks (Lin et al., 2016; Glass et al., 2018). However, even with this technique, approaching RE as a classification task presents several limitations: (1) distant supervision models are not accurate in extracting relations with a long-tailed distribution, because they typically have a small set of instances in knowledge bases; (2) in most domains, relation types are very specific and only a few examples of each relation are available; (3) these models cannot be applied to recognize new relation types not observed during training.

In this paper, we address RE from a different perspective by reducing it to an analogy problem. Our assumption states that if two pairs of entities, (A, B) and (C, D), have at least one relation in common r, then those two pairs are analogous. Viceversa, solving proportional analogies, such as $A : B = C : D$, consists of identifying the implicit relations shared between two pairs of entities. For example, ROME:ITALY=PARIS:FRANCE is a valid analogy because `capitalOf` is a relation in common.

Based on this idea, we propose an end-to-end neural model able to measure the degree of analogical similarity between two entity pairs, instead of predicting a confidence score for each relation type. An entity pair is represented through its

Proceedings of NAACL-HLT 2019, pages 3235–3245
Minneapolis, Minnesota, June 2 - June 7, 2019.

mentions in a textual corpus, sequences of sentences where entities in the pair co-occur. If a mention represents a specific relation type, then this relationship is expressed by the linguistic context surrounding the two entities. E.g., "Rome is the capital of Italy" or "The capital of France is Paris" referring to the example above. Thus, given two analogous entity pairs represented by their textual mentions sets as input, the model is trained to minimize the difference between the representations of relations having the same linguistic patterns. In other words, the model learns the different paraphrases expressing the same relation. In our research hypothesis, a model trained in such way is able to recognize analogies between unseen entity pairs belonging to new unseen relation types by: (1) generalizing over the sequence of words in the mentions; (2) projecting the sequence of words in the mentions into a vector space representing relational semantics. This approach poses several research questions: (RQ1) How to collect and organize a dataset for training? (RQ2) What kinds of models are effective for this task? (RQ3) How should the model be evaluated?

Knowledge bases, such as Wikidata or DBpedia, consist of large relational data sources organized in the form of triples, `predicate`(SUBJECT, OBJECT). We exploit this information to build a reliable set of analogous facts used as ground truth. Then, we adopt distant supervision to retrieve relation mentions in web-scale textual corpora by matching the subject-object entities which co-occur in the same sentences (Riedel et al., 2010; ElSahar et al., 2018; Glass and Gliozzo, 2018a). Through this technique we can train our model on millions of analogy examples without human supervision.

Since our goal is to train a model able to compute the relational similarity given two sets of textual mentions, we use siamese networks to learn discriminative features between those two instances (Hadsell et al., 2006). This kind of neural network has been used in both computer vision (Koch et al., 2015) and natural language processing (Mueller and Thyagarajan, 2016; Neculoiu et al., 2016) in order to map two similar instances close in a feature space. However, in our setting each instance consists of a set of mentions, therefore it is inherently a multi-instance learning task[1]. We propose a hierarchical siamese network with an attention mechanism at both word level (Yang et al., 2016) and at the set level (Ilse et al., 2018) in order to select the textual mention which better describes the relation. To the best of our knowledge, this is the first application of a siamese network by pairing sets of instances, so it can be considered a novelty of this work.

We evaluate the generalization capability of our model in recognizing unseen relation types through an one-shot relational classification task introduced in this paper. We train the parameters of the model on a subset of most frequent relations of one of three different distantly supervised datasets used in our experiments. Then, we evaluate it on the long-tailed relations of each dataset. During the test phase, only a single example for each unseen relation is provided. This example is not used to update the parameters of the model as in a classification task, but rather to produce the vector representation of the relation itself. Entity pairs having mention sets close to this representation are more likely to be analogous. The experiments show promising results of our approach on this task, compared with the recent deep models commonly used for encoding textual representations (Conneau et al., 2017). However, when the number of the unseen relation types increases, the performance of our model become far from the results obtained in the one-shot image classification (Koch et al., 2015), opening an interesting challenge for future work.

Finally, our model shows a transfer capability in other tasks through the use of its pre-trained vectors. Indeed, a branch of the hierarchical siamese network can be used to generate entity-entity representations given sets of mentions as input, that we call *analogy embeddings*. In our experiments, we integrate those representations into an existing end-to-end model based on convolutional networks (Glass and Gliozzo, 2018b), outperfoming the state-of-the-art systems on two shared datasets commonly used for distantly supervised relation extraction.

2 Related Work

Relation Extraction Several approaches have been proposed in the literature to address the problem of extracting relations from text with minimal supervision.

The bootstrapping method (Agichtein and Gra-

[1]Due to the weak supervision, the whole set of mentions is labeled, but each individual mention in the set is unlabeled.

vano, 2000) collects the textual patterns between a few example pairs of entities iteratively, and uses them to retrieve other pairs of entities from a corpus. This method is limited by the semantic drift issue since wrong patterns might be collected.

OpenIE (Mausam et al., 2012) is an unsupervised method for extracting triples from text, where the relations are linguistic phrases. The lack of a canonical form for the extracted relations makes this approach not suitable to populate knowledge bases with a fixed schema.

Universal schema (Riedel et al., 2013) addresses RE by combining the OpenIE and knowledge base relations through a matrix factorization technique typically adopted in the collaborative filtering approach of recommendation systems. The column-less (Toutanova et al., 2015) and row-less (Verga and McCallum, 2016) extensions of this method can handle unseen entity pairs and textual relations when combined (Verga et al., 2017).

The one-shot RE has been addressed by (Yuan et al., 2017), who adopt a siamese network to extract fine-grained relations which typically have few training examples. This model has two main limitations. Firstly, it works only by pairing two single mentions and is not able to handle a whole set of mentions referring to a relation instance. Our hierarchical siamese network overcomes this issue by using an attention mechanism at both word and mention level. Moreover, their one-shot evaluation mainly focuses on extracting the same relation types seen during training. Instead, the goal of our one-shot task is to evaluate the transfer capability in extracting unseen relation types across domains using a single pre-trained model.

Recently, (Levy et al., 2017) propose to reduce RE slot-filling to a question answering problem. The main idea is to build a set of question-answer pairs for the relations in knowledge bases and train a reading comprehension model using this dataset. This approach shows promising zero-shot capability in extracting unseen relation types. However, the schema querification phase requires a crowdsourcing effort. Our method uses distant supervision, so it does not need any kind of manual annotations.

Word Analogy The analogy problem, from a computational linguistic perspective, was originally addressed by (Turney, 2006) who investigate several similarity measures for solving word analogy questions in the Scholastic Aptitude Test dataset. The authors provide an interesting argument regarding the different types of similarities, attributional and relational, and their use in solving word analogies. Attributional similarity, typical of the word vector space models, is useful for synonym detection, word sense disambiguation and so on. Instead, relational similarity is suitable for understanding analogies between two pairs of words. Our neural-based analogy approach is inspired by this finding.

Recently, word analogies, namely the proportional analogy between two word pairs such as $a : b = c : d$, have been used by (Mikolov et al., 2013; Pennington et al., 2014) to show the capability of word embeddings to discover linguistic regularities in word contexts using vector offsets (e.g. $king - man + woman = queen$). The works in (Gladkova et al., 2016; Vylomova et al., 2016) explore the use of word vectors to model the semantic relations. The proportional analogy is also adopted by (Liu et al., 2017) as analogical inference in order to learn multi-relational embeddings which are evaluated on knowledge base completion benchmarks.

However, in order to apply word embedding models to proportional analogy, the model must have seen the words during training. This approach is unsuitable for computing the analogy between out-of-vocabulary words. Our approach overcomes this limitation, since it works by considering the contexts where the entities occur.

3 Learning Relations by Analogy

Given two pairs of entities, (A, B) and (C, D), their semantic relations can be expressed by their mentions in text, $(A, B) = \{S_i\}$ and $(C, D) = \{S_j\}$. Specifically, $\{S_i\}$ and $\{S_j\}$ are the sets of sentences where (A, B) and (C, D) co-occur in the same set. Two pairs of entities are analogous, $A : B = C : D$, iff their mentions sets, or part of them, express the same relation r. Knowledge bases, such as Wikidata, contain millions of trusted facts in form of triples, $r(A, B)$, namely pairs of entities in known relationships. We leverage these relational data sources as ground truth in order to collect a set of proportional analogy statements. Then, we build a dataset through the distant supervision technique by retrieving the mentions sets from web-scale corpora.

Our idea is to train a neural network to solve

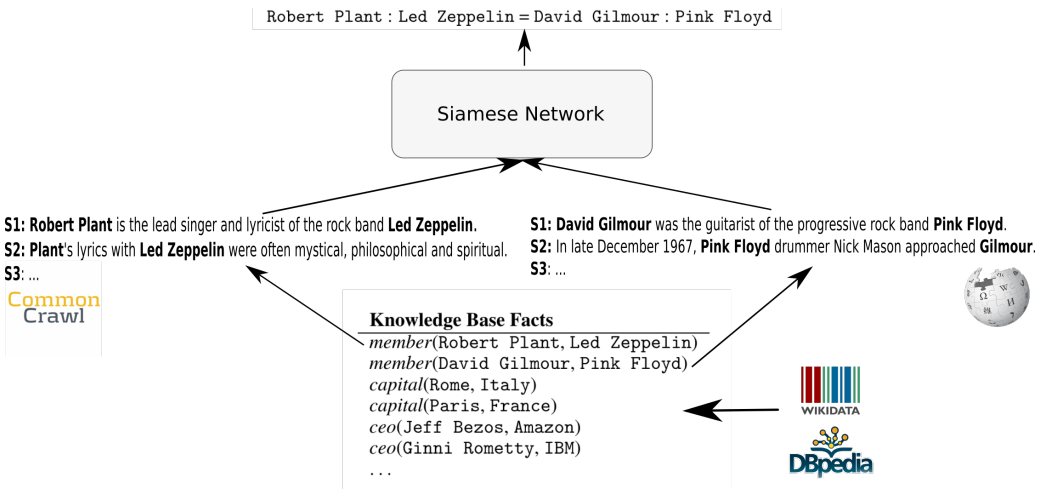


Figure 1: Learning relations by analogy through matching the facts from knowledge bases with textual corpora.

the analogy problem between any two entity pairs in this dataset, as long as they are described by the textual contexts where they co-occur. In other words, this task is reduced to a binary classification of determining whether the relational similarity between the representations of two sets of mentions exceeds a threshold. The network is trained by feeding two sets of mentions related to two different pairs, and it is optimized to return a positive label if the two entity pairs are analogous, namely they share at least one relation, or a negative label otherwise.

Figure 1 provides an example of this process. For the relation `memberOf`, the entity pairs (ROBERT PLANT, LED ZEPPELIN) and (DAVID GILMOUR , PINK FLOYD) are sampled. The two entity pairs are converted into their respectively mentions sets gathered from a textual corpus, such as Wikipedia. Since these two entity pairs are analogous, the network is optimized to learn the representations of the two textual contexts to be close into the feature space. In fact, the first sentences of both pairs represent the concept of membership of a band, even if they are expressed using different words. Based on our assumption, the aim is to learn how to encode the relational representations through the different paraphrases of the same relation. However, the model also needs negative examples during the training phase. We randomly select an entity pair from a different relation for each positive example, such as (ROBERT PLANT, LED ZEPPELIN) and (PARIS, FRANCE). Since we cast the problem as a binary classification task, we create a balanced dataset of positive and negative examples.

Siamese neural networks (Bromley et al., 1993) are well suited to this task because they are specifically designed to compute the similarity between two instances. A siamese network has symmetric twin sub-networks which share the same parameters, but are joined by an energy function at the head. Weight sharing forces the two similar instances to be mapped to very close locations in feature space because both of the sub-networks are optimized using the same function. In computer vision, siamese architectures based on convolutional neural networks (Hadsell et al., 2006; Koch et al., 2015; Vinyals et al., 2016) have shown promising performance in learning highly discriminative features by pairing images that belong to the same class. Likewise, our hypothesis is that a siamese network trained by matching two distinct mentions sets that share the same relation is able to learn how to map patterns of words across the sentences containing the two pairs of entities so as to capture the semantics of the relation. For instance, given the example in Figure 1 an effective siamese network should determine that the patterns for *"is the lead singer"* and *"was the guitarist"* express the same relation, `memberOf`.

To train a siamese network based on our approach, we have to face the following challenges: (1) the language may be highly variable and the same relation expressed in a multitude of different ways; (2) the mentions set of an entity pair consists of several sentences, each of which might express different relations, hence this is a multi-instance learning problem; (3) distant supervision could provide a wrong labeling, namely sentences which do not express any specific relations.

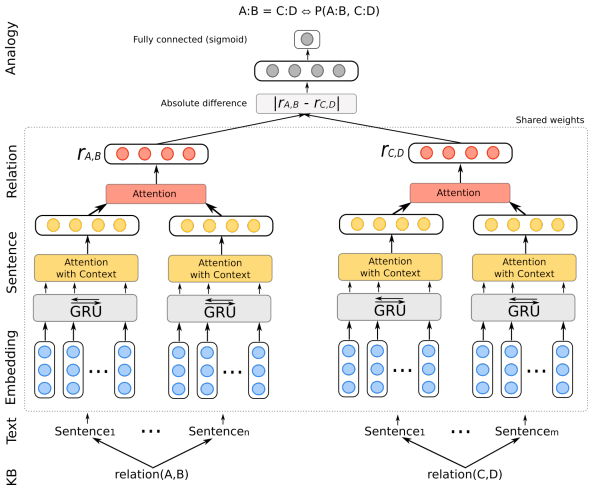


Figure 2: Hierarchical siamese network.

4 Hierarchical Siamese Network

To face these challenges, we propose a Hierarchical Siamese Network (HSN) architecture as shown in Figure 2. In the following paragraphs, we describe the details of each component.

Input Representation The HSN takes as input two entity pairs represented by their mentions sets. Since the twin sub-networks of the HSN are the same, we focus only on one of these. Given a triple $r(A, B)$ from a knowledge base, the relation r can be expressed through the set of sentences in a textual corpus where the two entities co-occur: $r(A, B) = \{S_1, S_2, \ldots, S_n\}$, with $S_i = \{w_{i1}, w_{i2}, \ldots, w_{ik}\}$, where w_{ij} represents the j-th word in the sentences S_i, $\forall i \in 1 \leq i \leq n$ and $\forall j \in 1 \leq i \leq k$. The purpose of a sub-network is to learn a low-rank vector representation $r_{A,B}$ for the relation r expressed by the pair (A, B). This is done by hierarchically composing the word and sentence representations.

Gated Recurrent Unit for Sentence Encoder Given a sentence $S_i = \{w_{i1}, w_{i2}, \ldots, w_{ik}\}$, we map each one-hot word representation of w_{ij} into its word embedding $x_{ij} = Ew_{ij}$, where $E^{d,|V|}$ is a matrix of real-valued vectors of size d, and V is a (fixed) vocabulary. Word embeddings are designed to encode syntactic and semantic features of words and can be randomly initialized or pre-trained on large corpora. We use pre-trained GloVe (Pennington et al., 2014) embeddings for our purposes. We encode the whole sentence S_i into a low-rank representation by composing its constituent word embeddings. An effective way to perform such an encoding is using recurrent neural networks (RNN) which are able to compose word embeddings by taking into account their positions in the sentence conditioned on the previous words. For our model, this capability is critical in order to detect sequences of words which express a particular relation, such as *"is the capital of"*. We use a bidirectional GRU (Bahdanau et al., 2014) to gather the information from both directions for words. Formally, given $\overrightarrow{h_{ij}} = \overrightarrow{GRU}(x_{ij})$ and $\overleftarrow{h_{ij}} = \overleftarrow{GRU}(x_{ij})$, the hidden state $h_{ij} = [\overrightarrow{h_{ij}}, \overleftarrow{h_{ij}}]$ is a new dense representation of w_{ij} which encodes also the information of the whole sentence.

Word Attention with Context Vector However, only certain words in a sentence express the semantics of a relation, therefore we need a strategy to automatically identify them during the training. For example, the words *"singer"* and *"guitarist"* at both sides of Figure 1 are good candidates to express the relation `member`. We use the attention mechanism with a context vector proposed in (Yang et al., 2016) to reward such words which are important to the meaning of a relation and then aggregate their information in the sentence representation. In detail, $s_i = \sum_k \alpha_{ik} h_{ik}$, where $\alpha_{ij} = \frac{exp(u_{ij}^T u_w)}{\sum_k exp(u_{ik}^T u_w)}$, and $u_{ij} = tanh(W_w h_{ij} + b_w)$. The vector s_i represents the sentence S_i and is computed as the weighted sum of the GRU-based word vectors h_{ij} using the normalized attention weights α_{ij}. The parameters for the attention mechanism are the weights and biases W_w, b_w and u_w, the context vector, a global fixed vector which, independently from a specific word, represents a kind of query which helps to inform what is the most informative word for each analogy. The context vector u_w essentially works like a memory mechanism, as described in (Sukhbaatar et al., 2015; Kumar et al., 2016).

Attention for Multi-instance Relation Representation Once all sentences in the mentions set are encoded, the aim of the last layer of a sub-network is to produce the vector $r_{A,B}$ which represents the pair (A, B). However, while weak supervision guarantees a large amount of training data without any human intervention, wrongly labeled sentences inevitably occur. For instance, the S2 of the pair on the right side in the Figure 1 does not express the relation `member` precisely, therefore a wrong bias could propagate during the training phase. This issue is typically addressed through

a multi-instance setting, where a model should identify the correct instance(s) from a bag. Recently, end-to-end neural architectures have been proposed to address this multi-instance classification problem (Wang et al., 2018; Feng and Zhou, 2017) by proposing several ways to aggregate the unlabeled instances, such as taking their average. Our goal is to have a model which is able to properly select the most relevant sentences by ascribing different weights to the encoded sentences. For this purpose, we adopt an attention mechanism at the sentence level. It is important to point out that the sentences in the mentions set do not have any temporal relationship, therefore we adapt the standard attention strategies as described in (Ilse et al., 2018). In detail, $r_{A,B} = \sum_i \alpha_i s_i$ is the embedding of the relation r given the pair (A, B), with $\alpha_i = \frac{exp(u_i^T u_s)}{\sum_k exp(u_k^T u_s)}$, $u_i = tanh(W_s s_i + b_s)$, where W_s and u_s are parameters.

Merging Layer and Training Strategy There are several ways to merge the output of the two sub-networks in order to learn the analogical similarity between them. For instance, (Hadsell et al., 2006) propose a contrastive loss with the aim of decreasing the distance between two instance representations. However, we adopt the strategy proposed in (Koch et al., 2015), in which the metric distance is induced by a fully-connected layer with a sigmoidal output unit on the absolute difference between the representations output by twin networks. Thus, given $r_{A,B}$ and $r_{C,D}$ the two relation embeddings which encode the whole mentions sets of the two entity pairs, we can compute the degree of analogy between them with $p = \sigma(W_r(|r_{A,B} - r_{C,D}|))$, where the parameters W_r measure the importance of each element of the difference vector, and they are learned in a end-to-end fashion, together with the relation representations. We build a training set by pairing the mentions sets of the entity-entity pairs from a KB, following the idea discussed in the next section. Thus, we can reduce the analogy task to a binary classification problem, so that $p = P((A, B), (C, D); \Theta)$ is equal to 1 if $A : B = C : D$, 0 otherwise. We learn Θ (all the parameters of HSN) using a gradient-based method which minimizes a cross-entropy loss function with the L2 regularization.

	NYT-FB	CC-DBP	T-REX
Knowledge Base	Freebase	DBpedia	Wikidata
Corpus	New York Times	Common Crawl	Wikipedia
# words	239,877	8,445,417	4,062,498
# entity pairs	375,846	6,876,913	6,413,452
# relations	57	298	685
avg. mentions	1.9	3.8	3.2
avg. sent. length	41	37	25

Table 1: Statistics of the distantly supervised datasets.

5 Experiments

Once the analogy model is trained, it has two different capabilities. First, the whole HSN architecture can be used as a binary classifier in order to infer if two entity pairs, expressed by their mentions sets, are analogous. Second, we can use its subnetwork before the merge layer as a feature extractor to generate entity-entity vectors given sets of sentences as input which can be used as pretrained analogy embeddings in other tasks. We evaluate our model on the one-shot relational classification and distantly supervised relation extraction benchmarks.

5.1 Datasets

In the entire experimental protocol we exploit three different datasets (see the supplemental material for details). **T-REX** (ElSahar et al., 2018) is a large scale alignment dataset between Wikipedia abstracts and Wikidata triples, having 685 unique relations. **NYT-FB** (Riedel et al., 2010) is a standard benchmark for distantly supervised relation extraction. The text of New York Times was processed with a named entity recognizer and the identified entities linked by name to Freebase. **CC-DBP** (Glass and Gliozzo, 2018a) is a web-scale KB population benchmark. It combines the text of Common Crawl with the entity-relation-entity triples from 298 frequent relations in DBpedia. Mentions of DBpedia entities are located in text by matching the preferred label. This task is similar to NYT-FB, but it has a much larger number of relations, triples and textual contexts. The statistics of the three datasets are summarized in Table 1. Aside from the difference in size and KB adopted, it worth noting also the difference in terms of corpus style of these datasets. For instance, T-REX has well-written textual mentions, because the sentences are extracted from Wikipedia. Conversely, CC-DBP and NYT-FB contain dirtier sentences which mean a high probability of incurring wrong labeling.

5.2 Training and Implementation Details

For both benchmarks, we use the same analogy model trained only once on a subset of the relations in T-REX. In detail, we discard all relations having less than 20 entity pairs, collecting 482 relations. We sort the relations by the number of instances, and we took the most frequent 60% of them to train the HSN. We use the remaining 20% of the relations for validation and the least frequent 20% as a corpus to implement one of the three one-shot classification tasks. For the validation and test partitions we randomly select only 20 entity pairs for each relation. This becomes a useful test set for the one-shot validation. To train the HSN, we select a balanced number of positive and negative examples out of the training split based on these rules: (1) for each relation, we randomly extract a set of 20 entity pairs; (2) out of this set, we generated all possibile combinations, $\binom{20}{2} = 190$, as positive pairing examples; (3) for each combination, we create a negative example by randomly selecting an entity pair from another relation. After these steps, we collect a bucket of 109,820 proportional analogy training examples.

We iterate this process throughout the training phase by selecting a different buckets at each iteration to prevent overfitting. The training is monitored by computing the binary accuracy over a fixed validation set, consisting of 36,480 analogy examples, built by adopting the same criteria described above. We initialize our word embedding layer with the pre-trained GloVe vectors consisting of 6B tokens with 50 dimensions. The word embedding weights are not updated during training. The number of mentions for each entity pair is fixed to 3, based on their average on T-REX (see Table 1).

5.3 One-shot Relational Classification

Task Given an unseen entity pair (A^t, B^t) and its mentions set $\langle A^t, B^t \rangle$, the one-shot relation classification task is to categorize this test pair (A^t, B^t) into one of N relation types, with the restriction that for each relation type $r_i, \forall i \in N$, we are given only one entity pair (A^i, B^i) together with its mentions set $\langle A^i, B^i \rangle$ as training. We can cast the one-shot classification in terms of a relational similarity as follows:

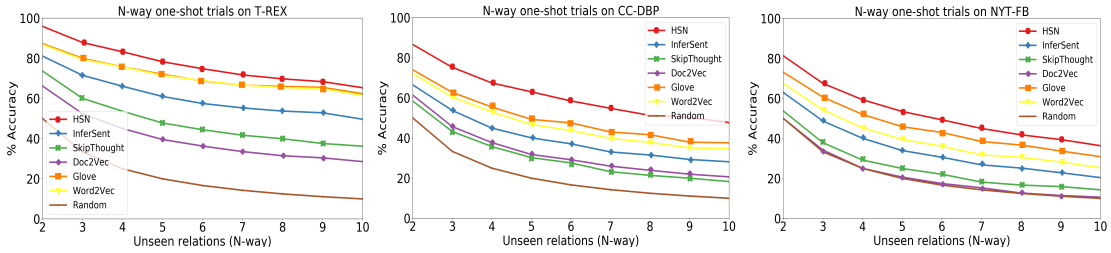
$$r_i = \arg\max_i sim_M(\langle A^t, B^t \rangle, \langle A^i, B^i \rangle) \quad (1)$$

where sim_M is a similarity score, using the method M, which measures the analogy between the train and test entity pairs through their mentions sets. We implemented sim_{HSN} using the HSN trained as described above. The two mentions sets are given as input to the network and their similarity is computed using the sigmoidal output of the last layer.

Baselines A method M should be robust in facing the *unseen* entity pairs used for testing. Training a standard RE model using just one example cannot provide a suitable baseline. Furthermore, since the two new entities that we want to classify might not be present in an existing knowledge graph, we could not apply relational embeddings (Bordes et al., 2013) as well. Thus, the use of the contexts surrounding the two entities in the mentions sets to compute the relational similarity score is needed. In other words, we cast the task of one-shot relational classification to a problem of measuring textual (i.e. mentions) similarity with the aim to prove that our pre-trained siamese model is able to grasp the semantics of relations better than the other pre-trained text representation models.

We implemented five baselines commonly used to encode textual representations. First, we use the pre-trained Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) embeddings. The score is given by the cosine similarity between the bag-of-means for the two entity pairs, averaging the word vectors in the mentions sets. We also adopt Doc2Vec (Le and Mikolov, 2014) to derive entity pair vectors, and comparing them using cosine similarity. For each entity pair, a pseudo-document embedding is created by concatenating its mentions sets. Finally, we compare HSM with the pre-trained Skip-Thought (Kiros et al., 2015) and InferSent (Conneau et al., 2017) sentence encoders, which are the state-of-the-art in computing textual similarity. An entity pair vector is obtained by averaging the embeddings of each sentence in the mentions set.

One-shot trials We follow the experimental setup described in (Koch et al., 2015) to create our one-shot benchmark. For each dataset, we select the 20% of less frequent relations sorted by the number of entity pairs, having at least 20 instances. Therefore, we collect three different one-shot test sets of 92, 55 and 11 unseen relation types for T-REX, CC-DBP and NYT-FB,



(a) T-REX (b) CC-DBP (c) NYT-FB

Figure 3: One-shot relational classification results for N-way unseen relation types.

respectively. The reason behind this criteria is to prevent the overlap of the semantic relation types between the train data and the different one-shot test sets. In fact, frequent relations, such as `location` or `birthPlace`, are common in all three datasets, and they are used to train our HSN. Moreover, using the long-tailed relations, such as `portOfRegistry`, we emulate a scenario where only a small set of relation examples are available, making more challenging the task. To evaluate the one-shot capabilities on N-way classes: (1) N different relation types are selected; (2) we sample one(shot) entity pair example for each of the selected N relation types; (3) we choose another entity pair used as test example from one of the N relation types. All the selections in these three steps are random. If the relation type returned by the Eq. 1 is equal to the relation type of the selected test example, then the one-shot trial is correct, otherwise it is incorrect. We repeated this operation k times for N from 2 to 10, for each of the three datasets. We choose k equal to 10,000, so that the random baseline converges to $100/N$, in order to create an unbiased testbed.

Results and Discussion The results are reported in Figure 3. Our model outperforms all the baselines on the test split of T-REX, reaching an accuracy range from 95.87% to 65.33% for N-way one-shot trials. This behavior remains constant also for the other two datasets, showing the solidity of HSN even though it has been trained on a different corpus using relations from an another ontology. This result confirms that our model is able to generalize on the linguistic contexts expressing relations, as well as the capability to learn how to transfer this information to other relations not observed before. The supplemental file reports some one-shot trial examples.

The lower accuracy on CC-DBP and NYT-FB might be caused by the different style of the corpora (Wikipedia vs. Web pages). Indeed, the test set of T-REX is build using the same corpus which HSN is trained on. The Wikipedia abstracts consist of well-written contents, typically the definition of one of the two entities in the pairs. Thus, T-REX can be considered an easier dataset compared with the other two.

Surprisingly, the average vectors using Word2Vec and GloVe obtain remarkable performance compared to state-of-the-art sentence encoders. This might be due to the way how these sentence models are trained. For instance, InferSent is trained using a natural language inference dataset, which might be not suitable to learn representations which represent relations in text. Instead, HSN is trained and optimized to learn and encode relational representations. However, this aspect deserves to be dealt with more deeply, as does the comparison of HSN on the shared textual similarity benchmarks; we think this is a clear path for future research.

5.4 Transfer Learning in Relation Extraction

We also evaluate the ability of the analogy model to provide low-rank representations for entity pairs which are useful for more traditional relation extraction tasks, where a corpus of text has to be processed and relevant relations in a predefined schema have to be recognized.

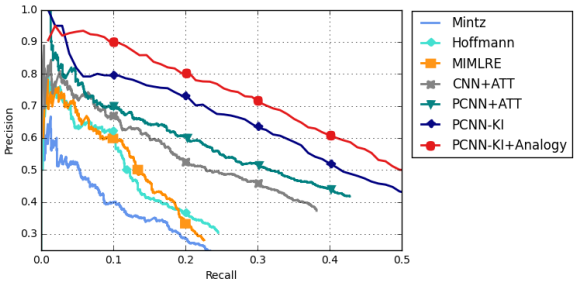


Figure 4: Precision-Recall curves on NYT-FB.

Relation / Score	Entity pair / Best mention	Entity pair / Best mention
doctoralStudent	VICTOR WEISSKOPF : MURRAY GELL-MANN	JOHN BARDEEN : NICK HOLONYAK
0.95	**Murray Gell-Mann**, one of the principal discoverers of the quarks, is one of the distinguished pupils of **Victor Weisskopf**.	Professor **Nick Holonyak** jr. was the first phd student of Nobel Prize winner **John Bardeen**.
approvedBy	HUNDRED HORSE CHESTNUT : GUINNESS WORLD RECORDS	NCSA OPEN SOURCE LICENSE : OPEN SOURCE INITIATIVE
0.83	**Guinness World Records** has listed **Hundred Horse Chestnut** for the record of "greatest tree girth eve".	**NCSA** was formally certified as an open-source license during a March 28, 2002 board meeting of the **Open Source Initiative**.
architecturalStyle	ROCKEFELLER CENTER : ART DECO	ST. MARK BASILICA : BYZANTINE ARCHITECTURE
0.63	**Art Deco** mural "wisdom" hangs over the entrance to the **Rockefeller Center** and was designed and sculpted by artist Lee Lawrie.	**St. Mark's Basilica**, the cathedral of Venice, is one of the best known examples of **Byzantine architecture**.

Table 2: Three examples of relational similarity between two pairs of entities computed by our HSN. For each example, we report the unseen relation type, the mentions related to each pair, and the similarity score. We report only the mention having the highest attention weight. The examples show the ability of the analogy model in providing a high score to two mentions which represent the same relation, even if they are expressed using different textual contexts.

To this aim, we use the sub-network of our HSN before the merge layer, and we feed the mentions set of each entity pair of instances as found in the corpus to generate an analogy embedding as a vector of features. In detail, given a set of mentions referring to an entity pair (A, B) as input, the pre-trained HSN generates an embedding $r_{A,B}$ (see Figure 2) which represents the relation between those two entities. Then, we concatenate these embeddings to the penultimate layer of a relation extraction model, PCNN-KI (Glass and Gliozzo, 2018b), based on a convolutional neural network, which is the state-of-the-art for this benchmark. The final fully-connected layer uses the representation from HSN in combination with its own learned multi-instanced vector representation to predict a confidence score for each relation. During the training of this joint model, PCNN-KI+ANALOGY, we freeze our analogy embeddings in order to avoid the loose the *knowledge transfer* capability.

As for the one-shot setting described before, we use the same pre-trained the HSN on the T-REX and we used it as a feature extractor for entity pairs in both train-test standard splits of NYT-FB, as used in (Zeng et al., 2015; Lin et al., 2016). Figure 4 reports the results of our evaluation. The model which uses the features generated by HSN largely improve the performances of PCNN-KI, despite the HSN is trained on a different corpus and using a different KB. In the same chart, we also report a compared evaluation for other approaches proposed in the literature for the NYT-FB benchmark: PCNN+ATT (Lin et al., 2016), CNN+ATT (Zeng et al., 2015), MIML-RE (Surdeanu et al., 2012), HOFFMANN (Hoffmann et al., 2011), MINTZ (Mintz et al., 2009).

We run the evaluation also on CC-DBP, a larger dataset for distantly supervised RE, using the same train-test setting adopted in (Glass and Gliozzo, 2018b). As done for the NYT-FB dataset, we incorporate the analogy embeddings generated by the same HSN trained on the T-REX. The results confirm the improvements obtained by PCNN-KI model if it integrates our pre-trained embeddings (Table 3).

	AUC	F1
PCNN-KI	0.437	0.468
PCNN-KI+ANALOGY	**0.500**	**0.504**

Table 3: AUC and F1 results on CC-DBP.

6 Conclusion and Future Work

In this paper, we proposed a novel approach to learn representations of relations in text. Alignments between knowledge bases and textual corpora are used as ground truth in order to collect a set of analogies between entity pairs. We designed a hierarchical siamese network trained to recognize those analogies. The experiments showed the two main advantages of our approach. First, the model can generalize on new unseen relation types, obtaining promising results in one-shot learning compared with the state-of-the-art sentence encoders. Second, the model can generate low-rank representations can help existing neural-based models designed for other tasks. As future work, we plan to continue our investigation by extending the method with other ideas. For instance, the use of positional embeddings, as well as the use of placeholders replacing the entities in the textual mentions are promising future directions. Finally, we plan also to explore the use of analogy embeddings in other tasks, such as question answering and knowledge base population.

References

Eugene Agichtein and Luis Gravano. 2000. *Snowball*: extracting relations from large plain-text collections. In *ACM DL*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *NIPS*.

Razvan C. Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *LREC*.

Ji Feng and Zhi-Hua Zhou. 2017. Deep MIML network. In *AAAI*.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *SRW@HLT-NAACL*.

Michael Glass and Alfio Gliozzo. 2018a. A dataset for web-scale knowledge base population. In *ESWC*.

Michael Glass and Alfio Gliozzo. 2018b. Discovering implicit knowledge with unary relations. In *ACL*.

Michael Glass, Alfio Gliozzo, Oktie Hassanzadeh, Nandana Mihindukulasooriya, and Gaetano Rossiello. 2018. Inducing implicit relations from text using distantly supervised deep nets. In *ISWC*.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *HLT*.

Maximilian Ilse, Jakub M. Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. *CoRR*, abs/1802.04712.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *ICML*.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *EMNLP-CoNLL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Rep4NLP@ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML PKDD*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL-HLT*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *NIPS*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP-CoNLL*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *EMNLP*.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3).

Patrick Verga and Andrew McCallum. 2016. Row-less universal schema. In *AKBC@NAACL-HLT*.

Patrick Verga, Andrew McCallum, and Arvind Neelakantan. 2017. Generalizing to unseen entities and entity pairs with row-less universal schema. In *EACL*.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NIPS*.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *ACL*.

Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. 2018. Revisiting multiple instance neural networks. *Pattern Recognition*, 74.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*.

Jianbo Yuan, Han Guo, Zhiwei Jin, Hongxia Jin, Xianchao Zhang, and Jiebo Luo. 2017. One-shot learning for fine-grained relation extraction via convolutional siamese neural network. In *BigData*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.