Data & Knowledge Engineering 117 (2018) 319–335

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Classification of composite semantic relations by a distributional-relational model

Siamak Barzegar[a,*], Brian Davis[b], Siegfried Handschuh[c], Andre Freitas[d]

[a] Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
[b] Department of Computer Science, Maynooth University, Ireland
[c] Department of Mathematics and Computer Science, University of Passau, Germany
[d] School of Computer Science, The University of Manchester, UK

ARTICLE INFO

Keywords:
Semantic relation
Distributional semantic
Deep learning
Classification

ABSTRACT

Different semantic interpretation tasks such as text entailment and question answering require the classification of semantic relations between terms or entities within text. However, in most cases, it is not possible to assign a direct semantic relation between entities/terms. This paper proposes an approach for composite semantic relation classification using one or more relations between entities/term mentions, extending the traditional semantic relation classification task. The proposed model is different from existing approaches which typically use machine learning models built over lexical and distributional word vector features in that is uses a combination of a large commonsense knowledge base of binary relations, a distributional navigational algorithm and sequence classification to provide a solution for the composite semantic relation classification problem. The proposed approach outperformed existing baselines with regard to F1-score, Accuracy, Precision and Recall.

1. Introduction

Capturing the semantic relationship between two concepts is a fundamental operation for many semantic interpretation tasks. This is a task which humans perform rapidly and reliably by using their linguistic and commonsense knowledge about entities and relations. Natural Language Processing (*NLP*) systems which aspire to reach the goal of producing meaningful representations of text must be equipped to identify and learn semantic relations in the documents they process. The automatic recognition of semantic relations has many applications such as information extraction, document summarization, machine translation, or the construction of thesauri and semantic networks. It can also facilitate auxiliary tasks such as word sense disambiguation, language modelling, paraphrasing, and recognizing textual entailment [25].

However, it is not always possible to establish a direct semantic relation given two entity mentions in text. In the *SemEval* 2010 Task 8 test collection [25] for example, 17.39% of the semantic relations mapped within sentences were assigned with the label *OTHER*, meaning that they could not be mapped to the set of 9 direct semantic relations.[1] In many cases, the semantic relations between two entities can only be expressed by a composition of two or more operations. For example, there is no direct relation

* Corresponding author.
E-mail addresses: siamak.barzegar@insight-centre.org (S. Barzegar), brian.davis@mu.ie (B. Davis), Siegfried.Handschuh@uni-passau.de (S. Handschuh), andre.freitas@manchester.ac.uk (A. Freitas).

[1] Cause-Effect, Instrument-Agency, Product-Producer, Content-Container, Entity-Origin, Entity-Destination, Component-Whole, Member-Collection, Communication-Topic.

https://doi.org/10.1016/j.datak.2018.06.005
Received 20 November 2017; Received in revised form 30 May 2018; Accepted 28 June 2018
Available online 04 July 2018
0169-023X/

between *Engineer* and *college* within current knowledge graph or semantic networks such as *DBpedia* or *ConceptNet*.

This work aims at improving the description and the formalization of the semantic relation classification task, in which the relations between entities can be expressed using the composition of one or more relations.

This paper is organized as follows: Section 2 describes the semantic relation classification problem and the related work (Section 3). Section 4.3 describes the existing baseline models, while Section 4 describes the experimental setup and analyses the results, providing a comparative analysis between the proposed model and the baselines. Finally, Section 5 offers a conclusion.

2. Composite semantic relation classification

2.1. Semantic relation classification

Semantic relation classification is the task of classifying the underlying abstract semantic relations between target entities (terms) present in texts [36]. The goal of relation classification is defined as follows: given a sentence S with pairs of annotated target nominals e_1 and e_2, the relation classification system aims to classify the relations between e_1 and e_2 in given texts within the pre-defined relation set [25]. For instance, the relation between the nominal ***burst*** and ***pressure*** in the following example sentence is interpreted as ***Cause-Effect***(e_2, e_1).

The $< e_1 >$ **burst** $< /e_1 >$ has been caused by water hammer $< e_2 >$ **pressure** $< /e_2 >$.

SemEval 2010 Task 8 [25] focuses on Multi-Way classification of semantic relations between pairs of nominals. For instance, *student* and *association* are in a ***Member-Collection*** relation in “*The student association is the voice of the undergraduate student population of the State University of New York at Buffalo*”.

They selected nine general relations plus “*OTHER*” is as follows:

- **Cause-Effect.** An event or object leads to an effect. *Example*: $< e_1 >$ *Smoking* $< /e_1 >$ causes $< e_2 >$ *cancer* $< /e_2 >$.
- **Instrument-Agency.** An agent uses an instrument. *Example*: $< e_1 >$ *Laser* $< e/_1 >$ $< e_2 >$ *printer* $< /e_2 >$
- **Product-Producer.** A producer causes a product to exist. *Example*: The $< e_1 >$ *farmer* $< /e_1 >$ grows $< e_2 >$ *apples* $< /e_2 >$
- **Content-Container.** An object is physically stored in a delineated area of space, the container. *Example*: $< e_1 >$ *Earth* $< /e_1 >$ is located in the $< e_2 >$ *MilkyWay* $< /e_2 >$.
- **Entity-Origin.** An entity is coming or is derived from an origin (e.g., position or material). *Example*: $< e_1 >$ *Letters* $< /e_1 >$ from $< e_2 >$ *foreigncountries* $< /e_2 >$.
- **Entity-Destination.** An entity is moving towards a destination. *Example*: The $< e_1 >$ *boy* $< /e_1 >$ went to $< e_2 >$ *bed* $< /e_2 >$.
- **Component-Whole.** An object is a component of a larger whole. *Example*: My $< e_1 >$ *apartment* $< /e_1 >$ has a large $< e_2 >$ *kitchen* $< /e_2 >$.
- **Member-Collection.** A member forms a nonfunctional part of a collection. *Example*: There are many $< e_1 >$ *trees* $< /e_1 >$ in the $< e_2 >$ *forest/* $< e_2 >$.
- **Communication-Topic.** An act of communication, whether written or spoken, is about a topic. *Example*: The $< e_1 >$ *lecture* $< /e_1 >$ was about $< e_2 >$ *semantics* $< /e_2 >$.

The final dataset contains a set of 10, 717 instances, where 8, 000 instances are defined as the training set. Table 1 shows the distribution of categories for the dataset. The second column (Frequency) shows the absolute and relative frequencies of each relation.

Table 1
Annotation Statistics of relation types with absolute and relative frequency in the dataset.

Relation	Frequency
Cause-Effect	1331 (12.4%)
Component-Whole	1253 (11.7%)
Entity-Destination	1137 (10.6%)
Entity-Origin	974 (9.1%)
Product-Producer	948 (8.8%)
Member-Collection	923 (8.6%)
Message-Topic	895 (8.4%)
Content-Container	732 (6.8%)
Instrument-Agency	660 (6.2%)
Other	1864 (17.4%)
Total	10717 (100%)

2.2. Existing approaches for single semantic relation classification

Different approaches have been explored for relation classification, including **unsupervised** and **supervised** relation discovery and classification. Existing literature has proposed various features to identify the relations between entities using different methods, which are described in the following paragraphs.

In the unsupervised methods, contextual features are used. The distributional hypothesis [23] indicates that words that have similar meanings, probably occur in the same context. Accordingly, it is assumed that the pairs of words that occur in similar contexts tend to have similar relations. Hasegawa et al. [24] used the contexts of nominal words using a hierarchical clustering method which represents the relationship between the words by using the most frequent words in the contexts. Chen et al. [9] suggested an unsupervised algorithm for addressing this problem based on model-order selection and discriminative label identification.

In the supervised methods, approaches can be grouped into two types: *feature-based* and *kernel-based* (see Ref. [50] for more details). The performance of these models strongly depends on the quality of the designed features. Recently, Neural network-based approaches have achieved significant improvement over traditional methods based on human-designed features [36]. Existing neural networks for relation classification are usually based on shallow architectures (e.g., one-layer convolutional neural networks or recurrent networks). In exploring the potential representation space at different abstraction levels, they may fail to perform [47]. The performance of supervised approaches strongly depends on the quality of the designed features [50]. Complementarily, some models are exploring automatic feature learning strategies. Xu et al. [48] introduce gated recurrent networks, in particular, Long Short-Term Memories (LSTMs) applied to relation classification. Zeng et al. [50] use Convolutional Neural Networks (CNNs) for the same task. Additionally, Santos et al. [38] replace the common Softmax loss function with a ranking loss in their CNN model. Xu et al. [46] design a negative sampling method based on CNNs. From the viewpoint of model ensembles, Liu et al. [30] combine CNNs and recursive networks along the Shortest Dependency Path (SDP), while Nguyen et al. [34] incorporate CNNs with Recurrent Neural Networks (RNNs).

Additionally, much effort has been invested in relational learning methods that can scale to large knowledge bases. The best performing neural-embedding models are Socher et al. (NTN) [41] and Bordes et al. models (TransE and TATEC) [8,21].

3. From single to composite relation classification

3.1. Introduction

The goal of this work is to propose an approach for semantic relation classification using one or more relations between entities/term mentions. In below example, the relationship between *Child* and *Cradle* cannot be directly expressed by one of the nine abstract semantic relations from the set described in Ref. [25].

The $< e_1 >$ *child* $< /e_1 >$ was carefully wrapped and bound into the $< e_2 >$ *cradle* $< /e_2 >$ by means of a cord.

Assume R_1 be a relation from X to Y, and R_2 be a relation from Y to Z. Then a relation written as $R_1 \bigcirc R_2$ is called a composite relation of R_1 and R_2 where

$$R_1 \circ R_2 = (x, y) | x \in X \wedge z \in Z \wedge (\exists\, y)(y \in Y \wedge (x, y) \in R_1 \wedge (y, z) \in R_2))$$

We can also write the composition as

$$R_1 \circ R_2 = (x, y) | x \in X \wedge z \in Z \wedge (\exists\, y)(y \in Y \wedge xR_1y \wedge yR_2z)$$

Note: Relational composition can be realized as matrix multiplication. For example, let M_{R_1} and M_{R_2} represent the binary relations R_1 and R_2, respectively. Then $R_1 \bigcirc R_2$ can be computed via $M_{R_1}\, M_{R_2}$.

Based on the definition of Composition of relation has been explained and looking into a commonsense KB (in this case, *ConceptNet V* 5.4) we can see the following set of composite relations between these elements:

$<e_1>$ *child* $</e_1>$ **CreatedBy∘Causes∘AtLocation** $<e_2>$ *cradle* $</e_2>$

With the increase in the number of edges which can be included in the set of semantic relation compositions (the size of the semantic relationship path), there is a dramatic increase in the number of paths which connect the two entities. For example, for the words *child* and *cradle* there are 15 paths of size 2, "1, 079" paths of size 3 and "95, 380" paths of size 4. Additionally, as the path size grows many non-relevant relationships (less meaningful or redundant relations) will be included.

The challenge in *composite semantic relation classification* is to provide a classification method that provides the most meaningful (see more details on Section 3.3) set of relations for the context at hand. This task can be challenging because, as previously mentioned, a simple KB lookup based approach would provide all semantic associations at hand. To achieve this goal we propose an approach which combines *sequence-based machine learning models*, *distributional semantic models* and *commonsense relational knowledge bases* to provide an accurate method for composite semantic relation classification. The proposed model (Fig. 1) relies on the combination of the following approaches:

i Using existing structured commonsense KBs to define an initial set of semantic relation compositions.
ii Using a pre-filtering method based on the Distributional Navigational Algorithm (DNA) as proposed by Refs. [20,39].
iii Using a sequence-based Neural Network based model to quantify the sequence probabilities of the semantic relation compositions. We call this model Neural Concept-Relation Model; an analogy to a Language Model.

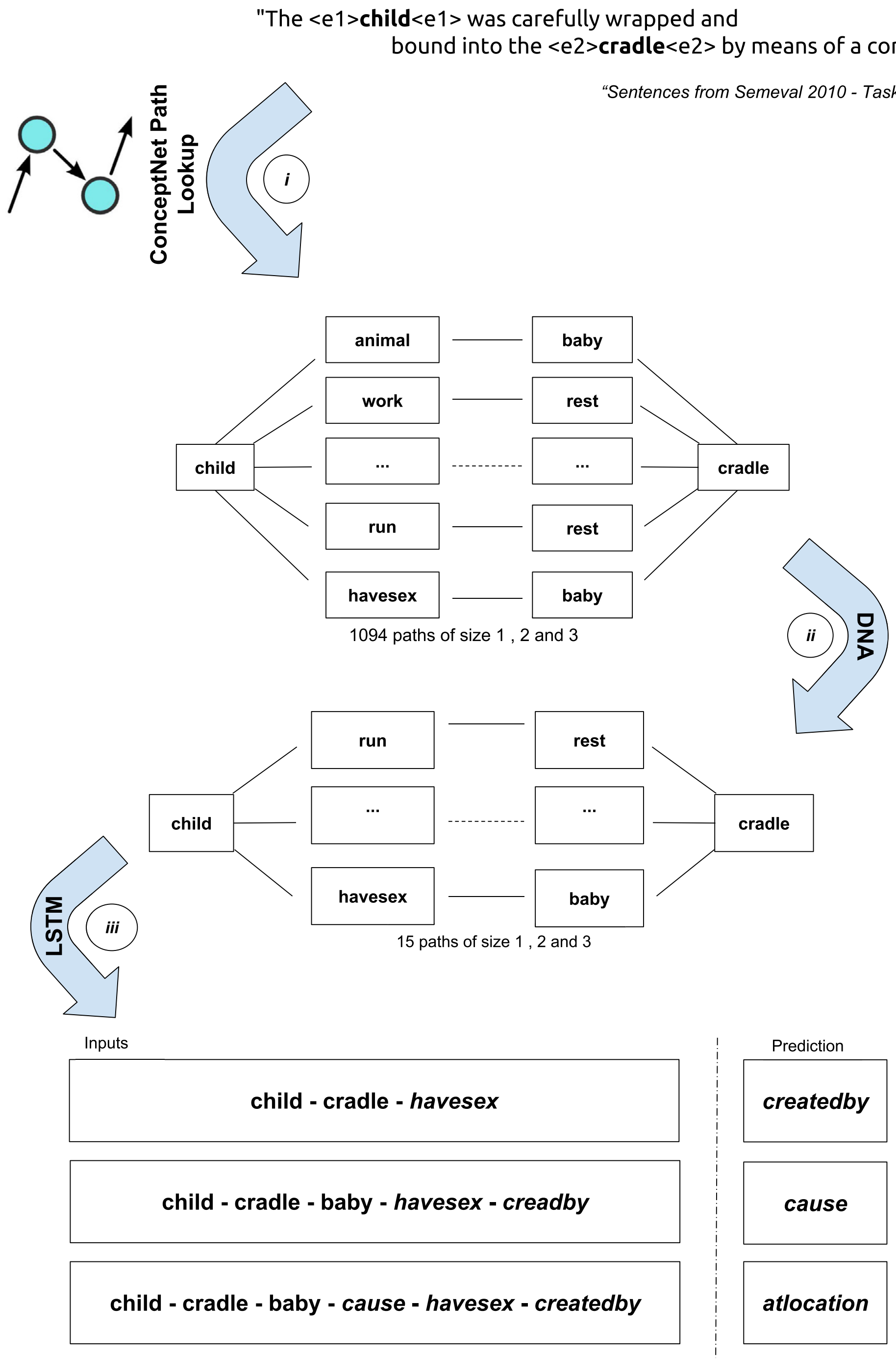

Fig. 1. Depiction of the proposed model relies on the combination of our three approaches.

3.2. Commonsense KB lookup

The first step consists in the use of a large commonsense knowledge base for providing a reference for a sequence of semantic relations. *ConceptNet* [44] is a semantic network built from existing linguistic resources and crowd-sourced. It is built from nodes representing words or short phrases as observed in natural language and labelled abstract relationships between them. There are a few alternative for large commonsense KBs, such as *WordNet*, *Microsoft Concept Graph* and *DBpedia*. In the list below, *ConceptNet* is contrasted to other KBs:

- **WordNet**: *ConceptNet* has more relation types than *WordNet*. Additionally, *ConceptNet*'s vocabulary[2] is much larger and contains more links between the Concepts. *ConceptNet* does not assume that words fall into *synsets*. Furthermore, *synonymy* in ConceptNet is a relation like any other. *ConceptNet* reuses some relations from WordNet. Moreover, their importance is weighted higher, given that the knowledge in *WordNet* is handcrafted, accurate and of high quality.
- **Microsoft Concept Graph**: *Microsoft Concept Graph* is a taxonomy of English nouns[3] containing *IsA* relations
- **DBpedia**: *DBpedia* is focused on named entities and basic factoid-style attributes. In contrast,*ConceptNet* focuses on noun/verb-level entities and their abstract relations.

ConceptNet is used as a large commonsense knowledge base for the proposed model. The intuition is that any type of relation classification task would need to be based on large-scale commonsense knowledge either in a distributional or structured/relational form. ConceptNet[4] has been built from several sources, such as:

- Information extracted from parsing Wiktionary [51]
- Open Multilingual WordNet [7]
- Open Mind Common Sense [40]
- A subset of DBpedia [1]

ConceptNet has a long-trail distribution of relations. However, the more frequent relations expressed at *ConceptNet* are[5]:

- **Symmetric relations:** *Antonym, DistinctFrom, EtymologicallyRelatedTo, LocatedNear, RelatedTo, SimilarTo, and Synonym.*
- **Asymmetric relations:** *AtLocation, CapableOf, Causes, CausesDesire, CreatedBy, DefinedAs, DerivedFrom, Desires, Entails, ExternalURL, FormOf, HasA, HasContext, HasFirstSubevent, HasLastSubevent, HasPrerequisite, HasProperty, InstanceOf, IsA, MadeOf, MannerOf, MotivatedByGoal, ObstructedBy, PartOf, ReceivesAction, SenseOf, SymbolOf, and UsedFor.*

For our target example, 1, 094 paths were extracted from *ConceptNet*for two given entities (e.g. *child* and *cradle*) such that they contained **no corresponding semantic relation** from the *SemEval* 2010 Task 8 test collection (Fig. 1(i)). Examples of paths are[6]:

- **child/CanBe/baby/AtLocation/cradle**
- child/IsA/animal/HasA/baby/AtLocation/cradle
- child/HasProperty/work/CausesDesire/rest/Synonym/cradle
- child/InstanceOf/person/Desires/baby/AtLocation/cradle
- child/DesireOf/run/CausesDesire/rest/Synonym/cradle
- **child/CreatedBy/havesex/Causes/baby/AtLocation/cradle**

Although ConceptNet can provide a large commonsense Knowledge Base, as we transcend immediate single relations, paths start to conceptually drift away from the *source* and *target* concepts. In order to filter these relations into a set of semantically relevant paths, we apply the Distributional Navigational Algorithm (DNA), described in the next section.

3.3. Distributional navigational algorithm (DNA)

The Distributional Navigational Algorithm (*DNA*) consists of an approach which uses distributional semantic models as a relevance-based heuristic for selecting relevant facts attached to a contextual query over a structured KB. DNA provides an abductive reasoning style mechanism which operates over Distributional-Relation Models [14,16–19], i.e. models which enrich structured logical (triple-style) KBs with word-embedding style information.

The DNA approach focuses on addressing the following problems: (i) providing a semantic selection mechanism for facts which are relevant and meaningful in a particular reasoning & querying context and (ii) allowing coping with information incompleteness in large KBs. The DNA model starts from the *source* entity and navigates through the KB, computing the distributional semantic relatedness between the set of lexical elements associated with neighbouring nodes in the graph and the *target* entity. The semantic relatedness function is defined as:

$$sr(\overrightarrow{p_1}, \overrightarrow{p_2}) = \cos(\theta) = \overrightarrow{p_1} \cdot \overrightarrow{p_2}$$

where *sr*: $VS^{dist} \times VS^{dist} \rightarrow [0, 1]$.

An important point to emphasize is the fact that the distributional semantic relatedness function is defined *over an external/independent corpus* (in contrast to many existing approaches which define the embeddings based on the KB). The DNA method is not

[2] 28 million statements.
[3] 5.4 million concepts.
[4] Version 5.5.
[5] https://github.com/commonsense/conceptnet5/wiki/Relations.
[6] Paths in bold are considered semantically relevant.

coupled to a specific distributional/word embedding model and can use different types of models.

A threshold $\eta \in [0, 1]$ can be used to establish the desired semantic relatedness between two vectors: $sr(\overrightarrow{p_1}, \overrightarrow{p_2}) > \eta$. The information provided by the semantic relatedness function *sr* is used to identify elements in the KB with a similar meaning from the reference corpus perspective. The threshold is calculated following the semantic differential approach proposed in Ref. [20]. Multiword phrases are handled by calculating the centroid between the concept vectors defined by each word in the Distributional Navigation Algorithm (DNA) (Algorithm 1) [20,39].

In summary, given two semantically related terms *source* and *target* wrt a threshold η, the algorithm finds all paths from *source* to *target*, with length *l*, formed by concepts semantically related to *target* wrt η.

The *source* term is the first element in all paths. From the set of paths to be explored (*ExplorePaths*), the *DNA* selects a path and expands it with all neighbours of the last term in the selected path that are semantically related *wrt* the threshold η and but do not appear in that path. The stop condition is $sr(target, target) = 1$ or when the maximum path length is reached.

The paths $p = < t_0, t_1, \ldots, t_l >$ (*where* $t_0 = source$ *and* $t_l = target$) found by *DNA* are ranked *(line 14 of code)* according to the following formula:

$$rank(p) = \sum_{i=0}^{l} sr(\overrightarrow{t_i}, \overrightarrow{target})$$

Algorithm 1 can be modified to use a heuristic that allows to expand only the paths for which the semantic relatedness between all the nodes in the path and the target term increases along the path. The differential in the semantic relatedness for two consecutive iterations is defined as $\Delta target(t_1, t_2) = sr(\overrightarrow{t_2}, \overrightarrow{target}) - sr(\overrightarrow{t_1}, \overrightarrow{target})$, for terms t_1, t_2 and *target*. This heuristic is implemented by including an extra test in the line 7 condition, i.e., $\Delta target(t_k, n) > 0$.

Algorithm 1
Distributional Navigational Algorithm.

η : *threshold*
$(source, target)$: *pair of terms such as that* $sr(\overrightarrow{source}, \overrightarrow{target}) > \eta$
l : *path length*
$RankedPaths$: *a set of ranked score paths* $< (t_0, \ldots, t_1), score >$ *such that* $t_0 = source$ *and* $t_1 = target$

$t_0 \leftarrow source$
$Paths \leftarrow 0$
$ExplorePaths \leftarrow [(< t_0 >, sr(\overrightarrow{t_0}, \overrightarrow{target})]$
while $ExplorePaths \neq 0$ **do**
  remove $(< t_0, \ldots, t_k >, sr(\overrightarrow{t_k}, \overrightarrow{target}))$ **from** $ExplorePaths$
  if $k < l - 1$ **then**
    for each $(n \in neighbours(t_k) : sr(\overrightarrow{n}, \overrightarrow{target}) > \eta$ *and* $n \notin \{ t_0, \ldots, t_k \})$ **do**
      append $(< t_0, \ldots, t_k, n >, sr(\overrightarrow{n}, \overrightarrow{target}))$ **to** $ExplorePaths$
    end for
  else if $k = l - 1$ **then**
      append $(< t_0, \ldots, t_k, target >, 1)$ **to** $Paths$
  end if
end while
$RankedPaths \leftarrow sort(Paths)$
return $RankedPaths$

In Ref. [20], DSMs are used as a complementary semantic layer to the relational model, which supports coping with semantic approximation and incompleteness. For large-scale and open domain commonsense reasoning scenarios, model completeness, and full materialization cannot be assumed. A commonsense KB would contain vast amounts of facts, and a complete inference over the entire KB would not scale to its size. Although several meaningful paths may exist between two entities, there are a large number of paths which are not meaningful in a specific context. For instance, the reasoning path which goes through path (1) at Fig. 2 is not related or relevant to the classification goal of the entity pairs (the relation between *Child* of human and *Cradle*) and should be eliminated by the application of the Distributional Navigation Algorithm (DNA) [20,39], which computes the distributional semantic relatedness between the entities and the intermediate entities in the knowledge base path as a measure of semantic coherence. In this case, the algorithm navigates from e_1 in the direction of e_2 in the Knowledge Base using distributional semantic relatedness between the target node e_2 and the intermediate nodes e_n as a heuristic method.

3.4. Neural Entity/Relation Model (NERM)

The Distributional Navigational Algorithm provides a pre-filtering of the relations maximizing the semantic relatedness

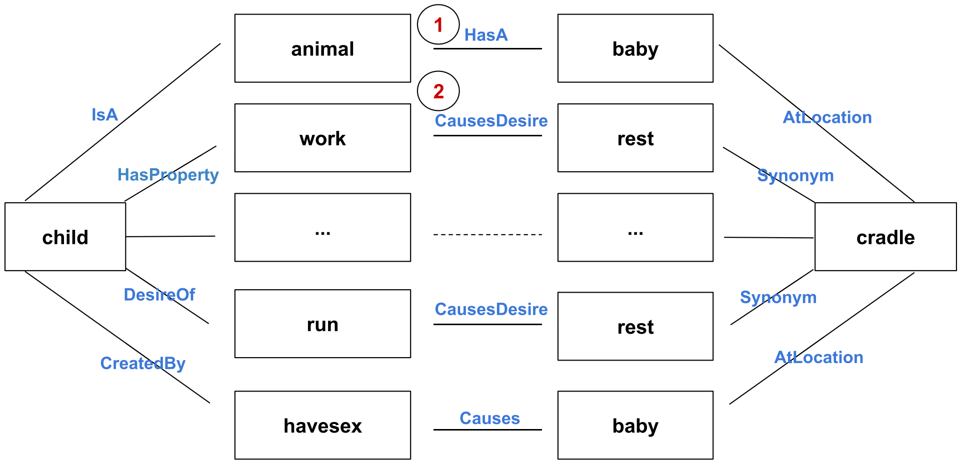

Fig. 2. Selection of semantically relevant paths.

coherence. This can be complemented by a predictive model which takes into account the likelihood of a sequence of relations, i.e. the likelihood of a composition sequence (Algorithm 2). The goal is to systematically compute the sequence of probabilities of entity-relation compositions, in a similar fashion to a language model. As such, the model will capture the notion of a sequence compatibility between entities and relations.

Various machine learning models are used in the context of NLP in order to induce language models and KB-based models, *Recursive Neural Networks* [42], *Recurrent Neural Networks* [11,32], *Long Short Term Memory Networks* [26], *Neural Tensor Networks* [41] and *Convolutional Neural Networks* [28]. In this work, due to the nature of the prediction task, which is similar to the language model (identifying semantically coherent sequences of relations), we will target recurrent sequence-classification types of models (see Fig. 4).

Algorithm 2
Composite Semantic Relation Classification.

I: *sentences of semeval 2010-Task 8 dataset*
O: *predefined entity pairs* (e_1, e_2)
W: *words in I*
R: *related relations of w*
for each $s \in I$ **do**:
  $S \leftarrow$ *If entities of s are connected in an OTHER relation*
end for
for each $s \in S$ **do**:
  $ep \leftarrow$ *predefined entity pairs of s*
  $p \leftarrow$ *find all path of ep in ConceptNet (with maximum paths of size 3)*
  for each $i \in p$ **do**:
    $sq_i \leftarrow$ *avg similarity score between each word pairs* [5]
  end for
  $msq \leftarrow$ *find* max sq
  for each $i \in p$ **do**:
    filter i If $sq_i < msq - \frac{msq}{2}$
  end for
  $dw \leftarrow$ *convert s into a suitable format for deep learning*
end for
$model \leftarrow$ *learning LSTM with dw dataset*

Long Short-Term Memory (LSTM) [26] is a type of recurrent neural network (*RNN*) architecture whose advantages over other RNN models have been proved in different tasks in NLP [4,10,45]. The advantage of *LSTM* against *RNN* is that allows the network to capture information from inputs for a long time using a special hidden unit (z_t).

An LSTM unit at a timestep t is described by the following equations:

$$i_t = \delta(W_{x,i}x_t + W_{z,i}z_{i-1})\ (input\ gate)$$

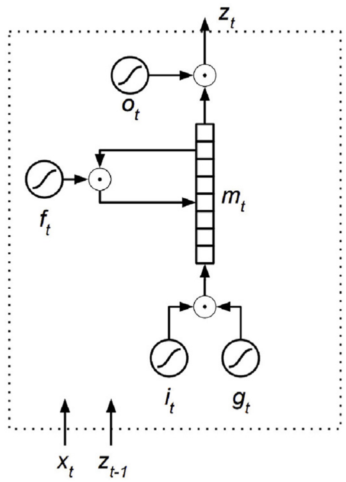

Fig. 3. Long Short-Term Memory unit at timestep *t*. (Small circles with dots are elementwise vector multiplications).

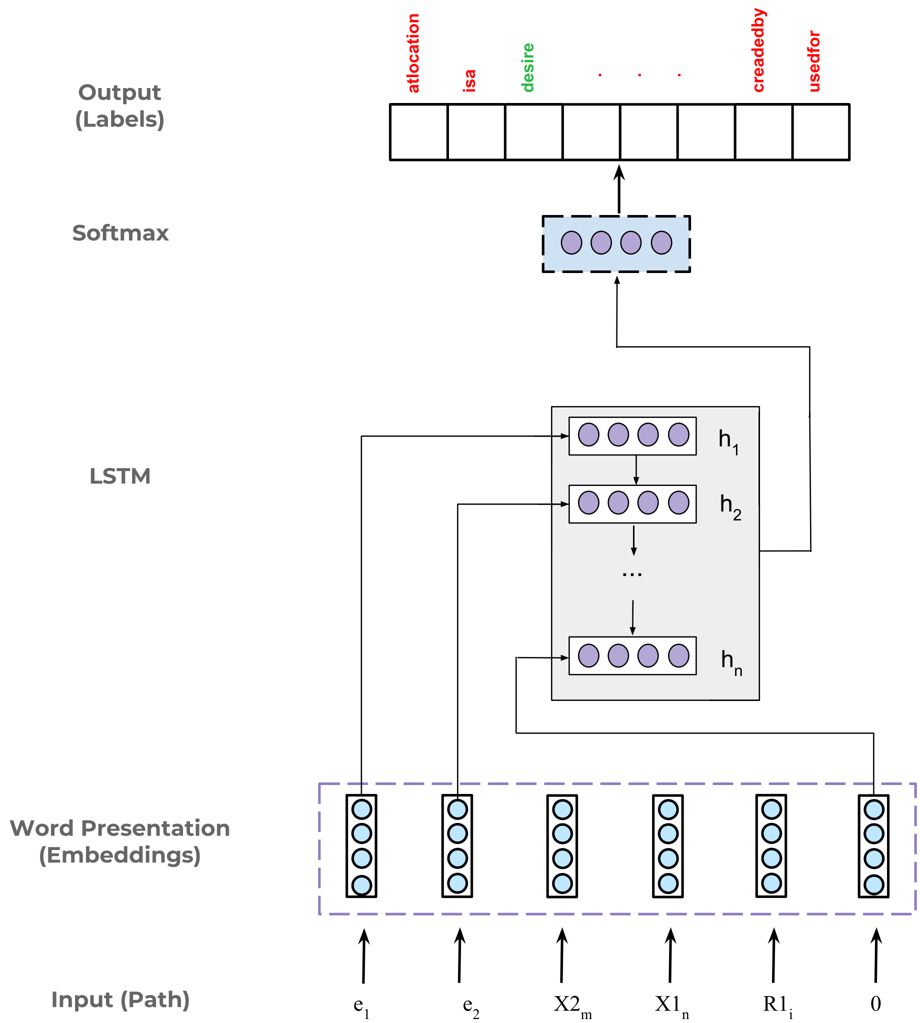

Fig. 4. The Neural Entity/Relation Model Architecture.

$$f_t = \delta(W_{x,f}x_t + W_{z,f}z_{f-1})\ (forgot\ gate)$$

$$o_t = \delta(W_{x,o}x_t + W_{z,o}z_{o-1})\ (output\ gate)$$

$$g_t = \tanh(W_{x,g}x_t + W_{z,g}z_{g-1})\ (input\ modulation\ gate)$$

$$m_t = f_t \circ m_{t-1} + i_t \circ g_t\ (memory\ cell)$$

$$z_t = o_t \circ \tanh(m_t)(hidden\ state)$$

The memory vector m_t is a function of two parts: (1) its previous value m_{t-1} modulated by the forgot gate f_t (2) the information of the current input x_t and previous hidden state (z_t) which modulated by the input modulation gate (g_t). Long Short-Term Memory unit at timestep t has four nonlinearity nodes (i_t, g_t, f_t, and o_t) all have, as inputs, x_t and z_{t-1} [29,35]. *LSTM* can memorize long sequences. The model's input is a sequence of entities and their relations with a specific order. For example, an input for our LSTM model is *child* − *cradle* − *baby* − *cause* − *havesex* − *creadtedby* which *model* should predict *atlocation* label.

The model is depicted graphically in Fig. 3.

4. Experimental evaluation

4.1. Training and test dataset

Two evaluation sets were generated by collecting all pairs of entity mentions in the *SemEval* 2010 task 8 [25]. The first dataset consists of entity pairs that have no attached semantic relation classes (i.e. which contained the relation label *OTHER*) while the second dataset contains *ALL* relations including relations labelled with *OTHER*.[7]

For all entities, we did a *ConceptNet* lookup [44], where we generated all paths from lengths 1, 2 and 3 (number of relations) occurring between both entities(e_1 and e_2) and their relations (R).

For example:

$$\mathbf{e1} - R1_i - \mathbf{e2}$$

$$\mathbf{e1} - R1_i - \mathbf{X1}_n - R2_j - \mathbf{e2}$$

$$\mathbf{e1} - R1_i - \mathbf{X1}_n - R2_j - \mathbf{X2}_m - R3_k - \mathbf{e2}$$

where X contains the intermediate entities between the target entity mentions **e1** and **e2**. Obviously, between two entities there may be different paths expressed with different intermediary entities and relations. For instance, for the paths between *silver* and *ring* entities we have:

- **silver**/UsedFor/**jewelry**/MadeOf/**gold**/AtLocation/**ring**
- **silver**/Antonym/**gold**/AtLocation/**ring**
- **silver**/Antonym/**bronze**/Antonym/**gold**/AtLocation/**ring**
- **silver**/AtLocation/**jewelry**/MadeOf/**gold**/AtLocation/**ring**

In next step, the *Distributional Navigational Algorithm (DNA)* is applied over the entity paths [20,39]. In the final step of generating the training & test datasets from 3, 728 entity pairs assigned to *OTHER* relation label in *SemEval* (*OTHER* dataset), we found 20, 261 paths and from 21, 434 entity pairs assigned to *ALL* relations in *SemEval* (*ALL* data set), we found 111, 526 paths in *ConceptNet*.

All paths were converted into the different formats with a specific order of entities and relations, which will be input for Neural Entity/Relation Model (Tables 2–4).[8] After converting to the new format for our Neural Entity/Relation Model (NERM), we find that we have 25, 260 and 141, 397 unique example relations for both datasets: *OTHER* and *ALL*, respectively.

4.2. Baseline models

The performance of baselines is measured using the test dataset, as defined in Section 4.1 where we hold out the last relation and rate a system by its ability to infer this relation.

As baselines, we use language models which define the conditional probabilities between a sequence of semantic relations $\boldsymbol{r}$ after the observation of entities e, i.e. $P(r|e)$.

- **Random Model:** This is the simplest baseline, which outputs randomly selected relation pairs.
- **Unigram Model:** Predicts the next relation based on unigram probability of each relation which was calculated from the training set. In this model, relations are assumed to occur independently.
- **Bigram Model:**

[7] Called *OTHER* and *ALL* sets, respectively.
[8] The best format based on our experiments is Table 2.

Table 2
First evaluation dataset for Neural Entity/Relation Model.

Input	Prediction
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{X1}_n$	$\mathbf{R1}_i$
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{X2}_m$ $\mathbf{X1}_n$ $\mathbf{R1}_i$	$\mathbf{R2}_i$
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{X2}_m$ $\mathbf{R2}_i$ $\mathbf{X1}_n$ $\mathbf{R1}_i$	$\mathbf{R3}_i$

Table 3
Second evaluation dataset for Neural Entity/Relation Model.

Input	Prediction
$\mathbf{e}_1$ $\mathbf{X1}_n$	$\mathbf{R1}_i$
$\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$ $\mathbf{X2}_m$	$\mathbf{R2}_i$
$\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$ $\mathbf{R2}_i$ $\mathbf{X2}_m$ $\mathbf{e}_2$	$\mathbf{R3}_i$

Table 4
Third evaluation dataset for Neural Entity/Relation Model.

Input	Prediction
$\mathbf{e}_2$ $\mathbf{e}_1$ $\mathbf{X1}_n$	$\mathbf{R1}_i$
$\mathbf{e}_2$ $\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$	$\mathbf{R2}_i$
$\mathbf{e}_2$ $\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$ $\mathbf{R2}_i$ $\mathbf{X2}_m$	$\mathbf{R3}_i$

The Bigram model is defined by Ref. [27]:

$$P(r|e) = \frac{P(r, e)}{P(e)} \tag{1}$$

where $P(r|e)$ is the probability of seeing e and r, in order. Let A be an ordered list of relations and entities, $|A|$ is the length of R, For $i = 1, .., |A|$, define a_i to be the ith element of A. We rank candidate relations r by maximizing P(r,a), defined as

$$P(r, a) = \sum_{i=1}^{|A|-1} \log P(r|a_i) \tag{2}$$

where the conditional probabilities $P(r|a_i)$ calculated using Equation (1).

- **Random Forest:** is an ensemble learning method for classification and other tasks, that operates by constructing a multitude of decision trees at training time. Random decision forests correct for the decision trees' limitation of overfitting to their training set.

4.3. Prediction task

The Neural Entity/Relation Model predicts composite relations between two given entities ($\mathbf{e}_1$ and $\mathbf{e}_2$). Given a sequence of source and target entities and a sequence of relations between them, the task consists of the prediction of the next relation X_i.

A semantically relevant path $\mathbf{e}_1 - R1_i - \mathbf{X1}_n - R2_j - \mathbf{X2}_m - R3_k - \mathbf{e}_2$ is converted into the following formats for the classification task (for different path lengths):

We provide statistics for the generated datasets in Table 5, whereby our dataset is divided into a training set and a test set with scale (80 − 20%). Also we used 20% of the training set for cross-validation. For the *OTHER* dataset, we have 16, 166 examples for training, 4, 042 for validation and 5, 952 for testing and for *ALL* set we have 90, 493 examples for training, 22, 624 for validation and 28, 280 for testing.

4.4. Word embeddings

The experimental setup consists of the instantiation of the W2V distributional semantic model. *Word2Vector (W2V)* [31] provides

Table 5
Distribution of instances used to train the LSTM model.

Dataset	#Train	#Dev	#Test
OTHER	16, 166	5, 052	4, 042
ALL	90, 493	28, 280	22, 624

an efficient implementation of the continuous bag-of-words and skip-gram models for computing vector representations of words. These representations can be subsequently used in many natural language processing applications and for further research. `INDRA` [15,37] provides a software infrastructure which facilitates the experimentation and customization of multilingual Word Embedding Models [2,3], allowing end-users and applications to consume and operate over multiple word embedding spaces as a service. In the experimental setup, we used `INDRA` as a service to get word embeddings for our classification model.

4.5. Results

To achieve the classification goal, we generated a Neural Entity/Relation Model for the composite relation classification task. In our experiments, a batch size 25, with 50 epochs was generated. An embedding layer was defined using Word2Vec pre-trained vectors.

In our experiment, we optimized the hyper-parameters of the LSTM model. After several preliminary experiments, the best model was achieved with the following set of parameters:

- Input length and dimension are 6 and 300, respectively.
- Three hidden layers with 450, 200 and 100 nodes and *Tanh* activation,
- Dropout technique (0.5),
- *Adam* optimizer.

We configured our Neural Entity/Relation Model and conducted experiments with three different pre-trained embedding settings:

- *Word2Vec (Google News)* with 300 dimensions
- *Word2Vec (Wikipedia 2016)* with 30 dimensions
- No pre-trained word embedding

The accuracy for the configuration above after 50 epochs is shown in Table 6. Table 7 contains the *Precision*, *Recall*, *F1-Score* and *Accuracy* metrics.

Table 7 shows the comparative analysis between NERM and the existing baselines.

Between the evaluated models, the Neural Entity/Relation Model achieved the highest F1 Score and Accuracy. The Bigram model achieved the second highest accuracy 0.3793 followed by Random forest model 0.3299. The LSTM approach provides an improvement of 9.86% on accuracy over the baselines, and 11.31% improvement on the F1-score. Random Forest achieved the highest precision, while the Entity/Relation Model achieved the highest recall.

The extracted information from the confusion matrix is shown in Tables 8 and 9. These two tables are calculated based on the first version of our evaluation, in which we have 3, 120 examples for training, 551 for validation and 1, 124 for testing. In Table 8 the '*Correctly Predicted*' column indicates the proportion of relations that are predicted correctly, and '*Correct Prediction Rate*' column indicates the rate at which the relations correctly predicted. For instance, our model predicts the relation '*NotIsA*' correctly in 100% of the cases.

Table 9 shows the relations which are wrongly predicted ('*Wrongly Predicted*' columns). Based on the results, the most incorrectly predicted relation is '*IsA*', which accounts for a large proportion of relations of the dataset (around 150 out of 550). In second place is the '*AtLocation*' relation (172 out of 550). In third place is the '*antonym*' relation. On the other hand, some relations which are not correctly predicted can be treated as semantically equivalent to their prediction, whereby their correct assignment depends on modeling decisions in the relation schema. The same situation occurs for specialization relations (e.g. '*EtymologicallyDerivedFrom*' and '*DerivedFrom*'). Another issue is the low occurrence of certain relations expressed in the dataset.

4.6. Enriching relationships

Based on the results at Table 9, some relations which are not correctly predicted can be treated as semantically equivalent to their prediction. Table 10 contains a description of a set of merged relations (merging more specific relations into more abstract categories with similar semantic function).

Also to keep the datasets coherent, we eliminate vague relations, such as ('*RelatedTo*'. '*DistinctFrom*', '*EtymologicallyRelatedTo*') and relations implying negation, such as ('*Antonym*' and '*Not*'). Table 11 shows the accuracy of CSRC(Composite Semantic Relation Classification) after merging relations that are semantically equivalent. Before merging similar relations, *OTHER* and *ALL* datasets contained 41 and 44 relations while after merging we have 18 and 21 relations, respectively.

Table 6
Validation accuracy.

CRSC	W2V Google News	W2V Wikipedia	No Pre-Training
Accuracy	0.4208	0.3841	0.2196

Table 7
Evaluation results on baseline models and our approach, with four metrics.

Method	Recall	Precision	F1-Score	Accuracy
Random	0.0160	0.0220	0.0144	0.0234
Unigram	0.0270	0.0043	0.0074	0.1606
Bigram	0.2613	0.2944	0.2502	0.3793
Random Forest	0.2476	0.3663	0.2766	0.3299
Entity/Relation model	0.3073	0.3281	0.3119	0.4208

Table 8
Extracted information from the Confusion Matrix - Part 1.

Relation	#Correct Predicted	Correct Predicted Rate
NotIsA	2	1
AtLocation	172	0.67
NotDesires	6	0.666
Similar	5	0.625
Desires	36	0.593
HasPrerequest	23	0.547
CausesDesire	17	0.548
IsA	147	0.492
Antonym	68	0.492
InstanceOf	46	0.479
UsedFor	47	0.475
DesireOf	5	0.5
HasContext	2	0.5
HasLastSubevent	2	0.5
NotHasA	1	0.5
MemberOf	1	0.5
HasA	24	0.393
HasSubEvent	12	0.378
PartOf	16	0.374
HasPropertry	12	0.375
Synonym	54	0.312
DerivedFrom	20	0.307
EtymologicallyDerivedFrom	6	0.3
CapableOf	13	0.26
MotivationByGoal	3	0.25
ReceiveAction	5	0.238
CreatedBy	4	0.2
MadeOf	3	0.16
Causes	3	0.15
Genre	1	0.11

4.7. Knowledge Base (KB) embeddings

The last part of the evaluation concentrates on assessing the impact of Knowledge Base (KB) embeddings into the ERM model.

4.7.1. Post-Processing Word Embeddings

Faruqui et al. [12] proposed a graph-based learning technique to obtain higher quality word embeddings by using lexical relational resources such as *Wordnet* [13], *Freebase* [6]. This technique known as *retrofitting*, brings semantically similar words close together while keeping them (relatively) close to their initial distributional vectors. It is a post-processing approach, whereby we inject semantic constraints into existing distributional vector spaces.

Speer et al. [43] introduced an ensemble method known as *ConceptNet-Numberbatch*, which combines data from pre-trained word embeddings and knowledge graphs, using a variation on retrofitting [12] to produce a high-quality word embeddings. They achieve this goal by applying the following method:

- Expanding the retrofitting algorithm [12] to benefit from structured links outside the original vocabulary.
- Using *ConceptNet* [44] as a resource of structured connections between words.
- Merging two pre-trained DSMs (Word2Vec and Glove) using a local linear interpolation. This combination performs better than each of the models separately.
- Applying expanded retrofitting method on the combined vector space model by using *ConceptNet* as a lexical relational resource.

Speer et al. called their word embedding *ConceptNet-Numberbatch*, showing that the combined embedding outperforms *W2V* on

Table 9
Extracted information from the Confusion Matrix - Part 2.

Relation	# Correct Predicted	Rate	Wrong Relation 1	# Falsely Predicted for Relation 1	Wrong Relation 2	# Falsely Predicted for Relation 2	Wrong Relation 3	# Falsely Predicted for Relation 3
AtLocation	172	0.67	Antonym	20	UsedFor	17		
Desire	36	0.593	IsA	6	CapableOf	6	UsedFor	5
HasPrerequest	23	0.547	Synonym	4	Antonym	3	AtLocation	2
CausesDesire	17	0.548	UsedFor	7				
IsA	147	0.492	AtLocation	26	Antonym	22	InstanceOf	22
Antonym	68	0.492	IsA	17	AtLocation	9		
InstanceOf	46	0.479	IsA	27	AtLocation	8		
UsedFor	47	0.475	AtLocation	26	IsA	18		
HasA	24	0.393	Antonym	11	UsedFor	6		
HasSubEvent	12	0.378	Causes	5	Antonym	4		
PartOf	16	0.374	Synonym	12	Antonym	3	HasProperty	3
HasProperty	12	0.375	IsA	8				
Synonym	54	0.312	IsA	31	HasProperty	17	AtLocation	12
DerivedFrom	20	0.307	IsA	10	Synonym	8	Etymologically-DerivedFrom	8
Etymologically-DerivedFrom	6	0.3	DerivedFrom	6				
CapableOf	13	0.26	UsedFor	13	IsA	7		
MotivatedByGoal	3	0.25	Causes	3	HasSubEvent	2		
ReceiveAction	5	0.238	AtLocation	9	UsedFor	3		
CreatedBy	4	0.2	Antonym	6	IsA	5		
MadeOf	3	0.16	IsA	7	Antonym	3	HasA	2
Causes	3	0.15	CausesDesire	6	HasSubEvent	4	DerivedFrom	3

Table 10
Merging similar relations with a more abstract relation.

Main Relation	Similar Relations
HasSubevent	HasFirstSubevent, HasLastSubevent, HasPrerequisite, Entails, MannerOf
Causes	MotivatedByGoal, CausesDesire
DerivedFrom	FormOf
SimilarTo	Synonym
IsA	InstanceOf, DefinedAs
LocatedNear	AtLocation, HasA, MadeOf, PartOf

Table 11
Accuracy before and after merging similar relations.

Word Embedding	Without Merging Relations *OTHER* Set	Merged Similar Relations *OTHER* Set	Merged Similar Relations *ALL* Set
W2V Google News	0.42	0.64	0.73

word-similarity evaluations.

In this work, we used *ConceptNet-Numberbatch* DSM as a pre-trained embedding variation instead of W2V model. Table 12 shows the accuracy of the CSRC classification using the *ConceptNet-Numberbatch* word embedding.

Speer et al. consider the data in ConceptNet as a symmetric matrix of association between words to apply the expanded retrofitting method. Therefore, they eliminate non-symmetric relations in ConceptNet and disregard these relation types to generate new word embeddings. We argue that in order to achieve a high quality semantic relation classification, all relations must be taken into account. Hence a more comprehensive approach is needed which includes knowledge about how both asymmetric and symmetric allowing us to inject all semantic constraints into existing word embeddings for completeness.

Table 12
Applying *ConceptNet-Numberbatch* as a pre-trained embedding vector space model in the CSRC classification model.

Word Embedding	Merged Similar Relations *OTHER* Set	Merged Similar Relations *ALL* Set
ConceptNet Numberbatch	0.66	0.74

Table 13
Use of CTransNet as a pre-trained embedding vector space model in the CSRC classification model.

Word Embedding	Merged Similar Relations *OTHER* Set	Merged Similar Relations *ALL* Set
CTransNet	0.73	0.80

Table 14
Comparison of accuracy scores of three types of Word Embeddings in our classification model (NERM).

Word Embedding	Without Merging Relations *OTHER* Set	Merged Similar Relations *OTHER* Set	Merged Similar Relations *ALL* Set
W2V - Google News	0.42	0.64	0.73
ConceptNet Numberbatch	N/A	0.66	0.74
CTransNet	N/A	0.73	0.80

4.7.2. Embedding entities and relations

As a second KB embedding model, we experimented with *translation embedding methods* as a pre-trained word embedding method.

Bordes et al. [8] proposed an energy-based model for learning low-dimensional embeddings of entities which is materialized into the *TransE* model. Relationships are represented as translations in the embedding space. In other words, the basic idea behind the model is, in a triple set (h, l, t) that composes two entities $h, t \in E$ the set of entities and a relationship $l \in L$(the set of relationships), the embedding of the entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship l.

$$h + l \simeq t$$

To learn such embeddings, they minimize a margin-based ranking criterion over the training set [49], where the scoring function of *TransE* is

$$-\,(2g_r^a(y_{e1}, y_{e2}) - 2g_r^b(y_{e1}, y_{e2}) + \| V_r \|_2^2)$$

where:

$$g_r^a(y_{e1}, y_{e2}) = A_r^T \begin{pmatrix} y_{e1} \\ y_{e2} \end{pmatrix} \quad and \quad g_r^b(y_{e1}, y_{e2}) = y_{e1}^T B_r y_{e2}$$

and A_r^T, B_r are relation-specific parameters and equal to $(V_r^T \quad - V_r^T)$ and I, respectively.

The main motivation of the translation-based parameterization is the structure of the hierarchical relationships that are very common in KBs; therefore translations are the best and natural transformations for representing them. Their model relies on a reduced set of parameters as it learns only one low-dimensional vector for each entity and each relationship. The optimization is carried out by stochastic gradient descent (using *minibatches*), and also the embedding vectors of the entities are normalized. *TransE* has fewer parameters compare with other approaches, leading to a simplification of the training process and preventing under-fitting.

A new word embedding called *CTransNet* was build by applying *STransE* [33] on the *ConceptNet* semantic network [44]. We trained *STransE* with *ConceptNet-Numberbatch* pre-trained word vectors, size = 300, l1 norm, margin = 5 and learning rate = 0.0005, nepoch = 2000 using *ConceptNet V* 5.5. Table 13 shows the accuracy of the CSRC classification using the *CTransNet* word embedding.

4.8. Final results

In the previous sections, we investigated the influence of different modalities of pre-trained models for the task of composite semantic relation classification in the context of the Neural Entity-Relation Model (NERM). Three models were analyzed: (1) traditional Word vector embeddings (W2V), (2) Post-Processing Word Embeddings and (3) Embedding Models of Entities and Relations.

The results (Table 14) shows that using *CTransNet* Word embedding outperforms the *W2V - Google News* and *ConceptNet-Numberbatch* word embedding on composite semantic relation classification task.

5. Conclusion

In this paper. we introduced the task of composite semantic relation classification. The paper proposes a composite semantic relation classification model which combines *commonsense KB lookup*, a *distributional semantics based filter* and the application of a *sequence-based machine learning model* to address the task.

The highest accuracy for the task of composite semantic relation classification was achieved by using Long-Short Term Memory as the sequence-based models and translation-based embeddings as the (Neural Entity-Relation Model). The proposed approach achieved 0.80 accuracy for the task at hand.

ConceptNet is built from nodes representing words or short phrases of natural language and *abstract* relationships between them. Future work will focus on enriching the relations with syntactic information. One example is the Syntactic Ngrams dataset, which contains dependency tree fragments extracted from of the Google Books corpus [22]. It contains a diverse set of relations, with maximal significance on relations between words. The dataset corpus is based on 3.5 million English books (Over 10 billion distinct items). A Syntactic Ngram is a rooted connected dependency tree over *n* words. For each *n* words in a sentence, a POS-tag[9] and basic dependency label for a given headword are provided. With this information, we can collect all SPO[10] relationships for each given word pairs for training our predict model. Also, we have a plan to compare our proposed LSTM model with other models such as CNN. Finally, additional baseline models such as SVM[11] will be also added in.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.datak.2018.06.005.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: a Nucleus for a Web of Open Data, The semantic web (2007), pp. 722–735.
[2] S. Barzegar, B. Davis, M.Z.S. Handschuh, A. Freitas. Semr-11: a Multi-lingual Gold-standard for Semantic Similarity and Relatedness for Eleven Languages.
[3] S. Barzegar, B. Davis, S. Handschuh, A. Freitas, Multilingual semantic relatedness using lightweight machine translation, Semantic Computing (ICSC), 2018 IEEE 12th International Conference on, IEEE, 2018, pp. 108–114.
[4] S. Barzegar, A. Freitas, S. Handschuh, B. Davis, Composite semantic relation classification, International Conference on Applications of Natural Language to Information Systems, Springer, 2017, pp. 406–417.
[5] S. Barzegar, J.E. Sales, A. Freitas, S. Handschuh, B. Davis, Dinfra: a one stop shop for computing multilingual semantic relatedness, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 1027–1028.
[6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, AcM, 2008, pp. 1247–1250.
[7] F. Bond, R. Foster, Linking and extending an open multilingual wordnet, ACL, 2013, pp. 1352–1362 (1).
[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.
[9] J. Chen, D. Ji, C.L. Tan, Z. Niu, Unsupervised feature selection for relation extraction, Proceedings of IJCNLP, 2005.
[10] K. Cortis, A. Freitas, T. Daudert, M. Huerlimann, M. Zarrouk, S. Handschuh, B. Davis, Semeval-2017 task 5: fine-grained sentiment analysis on financial microblogs and news, Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 519–535.
[11] J.L. Elman, Finding structure in time, Cognit. Sci. 14 (2) (1990) 179–211.
[12] M. Faruqui, J. Dodge, S.K. Jauhar, C. Dyer, E. Hovy, N.A. Smith, Retrofitting Word Vectors to Semantic Lexicons, arXiv preprint arXiv:1411.4166 (2014).
[13] C. Fellbaum, Wordnet and Wordnets, (2005).
[14] A. Freitas, Schema-agnositc Queries over Large-schema Databases: a Distributional Semantics Approach (Ph.D. thesis), (2015).
[15] A. Freitas, S. Barzegar, J.E. Sales, S. Handschuh, B. Davis, Semantic relatedness for all (languages): a comparative analysis of multilingual semantic relatedness using machine translation, Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings, vol. 20, Springer, 2016, pp. 212–222.
[16] A. Freitas, E. Curry, S. Handschuh, Towards a distributional semantic web stack, URSW, Citeseer, 2014, pp. 49–52.
[17] A. Freitas, J.C. Da Silva, S. O'Riain, E. Curry, Distributional relational networks, AAAI Fall Symposium Series, 2013.
[18] A. Freitas, S. Handschuh, E. Curry, Distributional-relational models: scalable semantics for databases, 2015 AAAI Spring Symposium Series, 2015.
[19] A. Freitas, J.C.P. da Silva, Semantics at Scale: when Distributional Semantics Meets Logic Programming, ALP Newsletter, 2014.
[20] A. Freitas, J.C.P. da Silva, E. Curry, P. Buitelaar, A distributional semantics approach for selective reasoning on commonsense graph knowledge bases, Natural Language Processing and Information Systems, Springer, 2014, pp. 21–32.
[21] A. Garcia-Duran, A. Bordes, N. Usunier, Y. Grandvalet, Combining two and three-way embedding models for link prediction in knowledge bases, J. Artif. Intell. Res. 55 (2016) 715–742.
[22] Y. Goldberg, J. Orwant, A dataset of syntactic-ngrams over time from a very large corpus of English books, * SEM@ NAACL-HLT, 2013, pp. 241–247.
[23] Z.S. Harris, Distributional structure, Word 10 (2–3) (1954) 146–162.
[24] T. Hasegawa, S. Sekine, R. Grishman, Discovering relations among named entities from large corpora, Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2004, p. 415.
[25] I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals, Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, Association for Computational Linguistics, 2009, pp. 94–99.
[26] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[27] B. Jans, S. Bethard, I. Vulić, M.F. Moens, Skip n-grams and ranking functions for predicting script events, Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2012, pp. 336–344.
[28] Y. Kim, Convolutional Neural Networks for Sentence Classification, arXiv preprint arXiv:1408.5882 (2014).
[29] Y.L. Kong, Q. Huang, C. Wang, J. Chen, J. Chen, D. He, Long short-term memory neural networks for online disturbance detection in satellite image time series, Rem. Sens. 10 (3) (2018) 452.
[30] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, H. Wang, A Dependency-based Neural Network for Relation Classification, arXiv preprint arXiv:1507.04646 (2015).
[31] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, arXiv preprint arXiv:1301.3781 (2013).
[32] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, Eleventh Annual Conference of the International Speech Communication Association, 2010.
[33] D.Q. Nguyen, K. Sirts, L. Qu, M. Johnson, Stranse: a Novel Embedding Model of Entities and Relationships in Knowledge Bases, arXiv preprint arXiv:1606.08140 (2016).
[34] T.H. Nguyen, R. Grishman, Combining Neural Networks and Log-linear Models to Improve Relation Extraction, arXiv preprint arXiv:1511.05926 (2015).
[35] K. Pichotta, R.J. Mooney, Using Sentence-level Lstm Language Models for Script Inference, arXiv preprint arXiv:1604.02993 (2016).
[36] P. Qin, W. Xu, J. Guo, An empirical convolutional neural network approach for semantic relation classification, Neurocomputing (2016).
[37] J.E. Sales, L. Souza, S. Barzegar, B. Davis, A. Freitas, S. Handschuh, Indra: a word embedding and semantic relatedness server, Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC, 2018.
[38] C.N. dos Santos, B. Xiang, B. Zhou, Classifying relations by ranking with convolutional neural networks, Proceedings of the 53rd Annual Meeting of the

[9] Penn-Treebank part of speech tag.
[10] Subject, Predicate, and Object.
[11] Support Vector Machines.

Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, 2015, pp. 626–634.
[39] V. Silva, S. Handschuh, A. Freitas, Recognizing and Justifying Text Entailment through Distributional Navigation on Definition Graphs, AAAI, 2018.
[40] P. Singh, et al., The public acquisition of commonsense knowledge, Proceedings of AAAI Spring Symposium: Acquiring (And Using) Linguistic (And World) Knowledge for Information Access, 2002.
[41] R. Socher, D. Chen, C.D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, Advances in Neural Information Processing Systems, 2013, pp. 926–934.
[42] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[43] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: an open multilingual graph of general knowledge, AAAI, 2017, pp. 4444–4451.
[44] R. Speer, C. Havasi, Representing general relational knowledge in conceptnet 5, LREC, 2012, pp. 3679–3686.
[45] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[46] K. Xu, Y. Feng, S. Huang, D. Zhao, Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling, arXiv preprint arXiv:1506.07650 (2015).
[47] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, Z. Jin, Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation, arXiv preprint arXiv:1601.03651 (2016).
[48] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, Z. Jin, Classifying relations via long short term memory networks along shortest dependency paths, Proceedings of Conference on Empirical Methods in Natural Language Processing (To Appear), 2015.
[49] B. Yang, W.t. Yih, X. He, J. Gao, L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, arXiv preprint arXiv:1412.6575 (2014).
[50] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, et al., Relation classification via convolutional deep neural network, COLING, 2014, pp. 2335–2344.
[51] T. Zesch, C. Müller, I. Gurevych, Extracting lexical semantic knowledge from wikipedia and wiktionary, LREC, vol. 8, 2008, pp. 1646–1652.

Siamak Barzegar is a final year PhD candidate at Insight Centre for Data Analytics, former Digital Enterprise Research Institute at the National University of Ireland, Galway under the supervision of Dr Andre Freitas and Dr Brian Davis. He won the Science Foundation Ireland (SFI) research scholarship to undertake his PhD in the area of Scalable Knowledge Extraction, Capturing and Discovery. The main area of his PhD research is focusing on generating a Distributional Semantics (Word Embedding) Architecture that is transportable across different domains and languages. During his PhD, for near two years, he was part of Prof. Handschuh's research group at the University of Passau (Germany) as a research placement. His research interests include Natural Language Processing, Distributional Semantics, Word Embeddings, Deep Learning, Machine Translation, Knowledge Extraction on Specific Domain. Before joining at the Insight Centre, Siamak worked at Telecommunication Infrastructure Company as a data analyst in fraud detection as a data analyst in fraud detection.

Brian Davis is currently a Lecturer(ATB) in Computer Science at Maynooth University, Co Kildare, Ireland. Prior to taking up this appointment, he was a Research Fellow, Adjunct Lecturer and Research Unit Leader at the INSIGHT Center for Data Analytics, NUIGalway (NUIG), where he led the Knowledge Discovery Unit focusing on the specific research areas of: Natural Language Processing, Data Visualization and Knowledge Discovery from heterogeneous data sources (text and graph). He was coordinator of a 3 year Horizon 2020 Innovation Action – SSIX - Social Sentiment Financial Indexes (Grant No. 645425). His core expertise intersects with Natural Language Processing, Ontology Engineering and Data Visualization. Other research interests include: NLP for social media, cross lingual opinion mining from social media for the finance and political domains and combining visualization and textualisation using Natural Language Generation. He has over eight years research experience in Ontology Based Information Extraction and Semantic Annotation.

Prof. Dr. Siegfried Handschuh is professor at Universität Passau, Fakultät für Informatik und Mathematik, Lehrstuhl für Informatik mit Schwerpunkt Digital Libraries and Web Information Systems. Prior to the acquisition of the Chair in Passau, he was professor at the National University of Ireland, Galway (NUIG) and head of the Knowledge Discovery Unit at the Insight Center for Data Analytics in Galway.

He studied computer science in Ulm and information science in Konstanz and was assistant at the Institute for Applied Computer Science and Formal Description Methods at the University of Karlsruhe, subsequently. In 2001, he continued with a research stay at Stanford Database Group at Stanford University in the United States. He received his doctorate with magna cum laude in Karlsruhe in 2005.

Prof. Handschuh coordinated numerous research and development projects on international level. He was involved in attracting extensive project funding: of the EU, the Science Foundation Ireland, Enterprise Ireland and from national sources, amongst others. In addition, he has worked with multinational companies such as HP, SAP, IBM, Motorola, Cisco, Avaya, British Telecom, Telecom Italia, Telefonica, Thales and Elsevier Publishing.

Furthermore, he conducted research at Digital Aristoteles Project that aimed at the semantic content analysis and knowledge modeling of schoolbooks, and was funded by the Microsoft founder and philanthropist, Paul Allen. His current research deals mainly with the areas of semantic technologies, information linguistics, information extraction and web information retrieval.

André Freitas is a lecturer (assistant professor) at the School of Computer Science at the University of Manchester. Prior to Manchester, he was an associate researcher and lecturer at the Natural Language Processing and Semantic Computing Group at the University of Passau (Germany) at the Chair of Digital Libraries and Web Information Systems. He is also a partner and co-founder at Amtera Semantic Technologies. Before joining Passau, he was part of the Digital Enterprise Research Institute (DERI) at the National University of Ireland, Galway where he did his PhD on Schema-agnostic Query Mechanisms for Large-Schema Databases. André holds a BSc. in Computer Science from the Federal University of Rio de Janeiro (UFRJ), Brazil (2005). His main research areas include Question Answering, Schema-agnostic Database Query Mechanisms, Natural Language Query Mechanisms over Large-Schema Databases, Distributional Semantics, Hybrid Symbolic-Distributional Models, Approximate Reasoning and Knowledge Graphs. Before joining DERI, André worked as a research assistant (trainee) at Siemens Corporate Research, Princeton, USA. André worked as a software engineer, product designer and project manager in different industries including Oil & Gas Exploration, IT Security, Medical, Healthcare, Banking, Mining and Telecom.