Analogy-based Reasoning With Memory Networks for Future Prediction

Daniel Andrade*
Data Science Research Laboratories
NEC Corporation, Japan
s-andrade@cj.jp.nec.com

Bing Bai
Department of Machine Learning
NEC Laboratories America
bbai@nec-labs.com

Ramkumar Rajendran†
Computer Science Department
Vanderbilt University
ramkumar.rajendran@vanderbilt.edu

Yotaro Watanabe
Data Science Research Laboratories
NEC Corporation, Japan
y-watanabe@fe.jp.nec.com

Abstract

Making predictions about what might happen in the future is important for reacting adequately in many situations. For example, observing that "Man kidnaps girl" may have the consequence that "Man kills girl". While this is part of common sense reasoning for humans, it is not obvious how machines can learn and generalize over such knowledge automatically. The order of event's textual occurrence in documents offers a clue to acquire such knowledge automatically. Here, we explore another clue, namely, logical and temporal relations of verbs from lexical resources. We argue that it is possible to generalize to unseen events, by using the entailment relation between two events expressed as (subject, verb, object) triples. We formulate our hypotheses of analogy-based reasoning for future prediction, and propose a memory network that incorporates our hypotheses. Our evaluation for predicting the next future event shows that the proposed model can be competitive to (deep) neural networks and rankSVM, while giving interpretable answers.

1 Introduction

Making predictions about what might happen in the future is important for reacting adequately in many situations. For example, observing that "Man kidnaps girl" may have the consequence that "Man kills girl". While this is part of common sense reasoning for humans, it is not obvious how machines can learn and generalize over such knowledge automatically.

One might think of learning such knowledge from massive amount of text data, such as news corpora. However, detecting temporal relations between events is still a difficult problem. Temporal order of events are often presented in different order in text. Although the problem can be partially addressed by using temporal markers like "afterwards", particularly with discourse parsers [18], overall, it remains a challenge.[3]

In this work, we propose to exploit the distinction between logical relations and temporal relations. We note that if an entailment relation holds between two events, then the second event is likely to be

*The first author is also associated with the Graduate University for Advanced Studies (SOKENDAI).

†The co-author contributed to this work while he was at NEC Corporation, Japan.

[3]For example, detecting implicit temporal relations (i.e. no temporal markers) is still a difficult problem for discourse parsers [18].

not a new future event.[4] For example, the phrase "man kissed woman" entails that "man met woman", where "man met woman" happens before (not after) "man kissed woman". To find such entailments, we can leverage relation of verbs in WordNet [5]. Verbs that tend to be in a temporal (happens-before) relation have been extracted on a large scale and are openly available in VerbOcean [4]. For example, we observe (subject, buy, object) tends to be temporally preceding (subject, use, object).

We present a model that can predict future events given a current event triplet (subject, verb, object). To make the model generalizable to unseen events, we adopt a deep learning structure such that the semantics of unseen events can be learned through word/event embeddings. We present a novel Memory Comparison Network (MCN) that can learn to compare and combine the similarity of input events to the event relations saved in memory. Our evaluation shows that this method is competitive to other (deep) neural networks and rankSVM [7], while giving interpretable answers.

In the first part of this work, in Section 2, we describe previous work related to future prediction. In Section 3, we discuss some connections between logical and temporal relations, and explain how we use lexical resources to create a knowledge base of positive and negative temporal relations. This knowledge base is then used by our experiments in the second part of our work.

In the second part, in Section 4, we formulate our assumptions of analogy based reasoning for future prediction. Underlying these assumptions, we propose our new method MCN. In Section 5, we describe several other methods that were previously proposed for future prediction, and ranking models that can be easily adapted to this task. In Section 6, we evaluate all methods on a future prediction task that requires to reason about unseen events. Finally, in Sections 7 and 8, we discuss some current limitations of our proposed method, and summarize our conclusions.

2 Related work

One line of research, pioneered by VerbOcean [4], extracts happens-before relations from large collections of texts using bootstrapping methods. In the context of script learning, corpora statistics, such as event bi-grams, are used to define a probability distribution over next possible future events [13, 3]. However, such models cannot generalize to situations of new events that have not been observed before. Therefore, the more recent methods proposed in [11, 15, 6] are based on word embeddings. Script learning is traditionally evaluated on small prototypical sequences that were manually created, or on event sequences that were automatically extracted from text. Due to the lack of training data, these models cannot learn to distinguish the fact that some events later in the text are actually entailed by events previously mentioned, i.e. already known events and new events are not distinguished.

3 Exploiting lexical resources

Our main focus is on distinguishing future events from other events. In texts, like news stories, an event e_l is more likely to have happened before event e_r (temporal order), if e_l occurs earlier in the text than e_r (textual order). However, there are also many situations where this is not the case: re-phrasing, introducing background knowledge, conclusions, etc. One obvious solution are discourse parsers. However, without explicit temporal markers, they suffer from low recall [18], and therefore in practice most script-learning systems use textual order as a proxy for temporal order. Here we explore whether common knowledge can help to improve future detection from event sequences in textual order.

We assume common knowledge is given in the form of simple relations (or rules) like

$$\text{(company, buy, share)} \rightarrow \text{(company, use, share)} ,$$

where "$\rightarrow$" denotes the temporal happens-before relation. In contrast, we denote the logical entailment (implication) relation by "$\Rightarrow$".

To extract such common knowledge rules we explore the use of the lexical resources WordNet and VerbOcean. As also partly mentioned in [5], logical and temporal relations are not independent, but

[4] We consider here entailment and (logical) implication as equivalent. In particular, synonyms are considered to be in an entailment relation, as in contrast to the classification by WordNet.

Table 1: Examples of several temporal and logical relations (relation types are shown in numbers relating to Figure 1).

Examples
(1) "minister leaves factory", "minister enters factory"
(2) "company donates money", "company gives money"
(3) "John starts marathon" , "John finishes marathon"
(4) "governor kisses girlfriend", "governor meets girlfriend"
(5) "people buy apple", "people use apple"
(6) "minister likes criticism", "minister hates criticism"
(7) "X's share falls 10%", "X's share rises 10%"

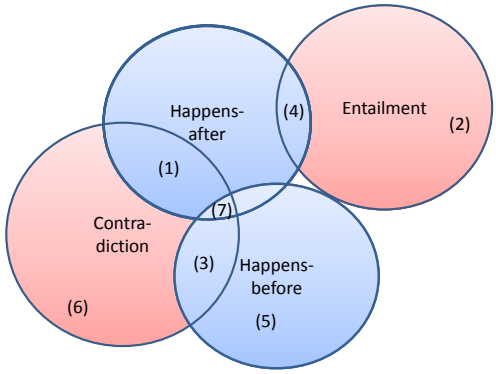

Figure 1: Illustration of logical (entailment, contradiction) and temporal (happens-before, happens-after) relation types. Examples are shown in Table 1.

an interesting overlap exists as illustrated in Figure 1, and corresponding examples shown in Table 1. We emphasis that, for temporal relations, the situation is not always as clear cut as shown in Figure 1 (e.g. repeated actions). Nevertheless, there is a tendency of event relations belonging mostly only to one relation. In particular, in the following, we consider "wrong" happens-before relations, as less likely to be true than "correct" happens-before relations.

3.1 Data creation

For simplicity, we restrict our investigation here to events of the form (subject, verb, object). All events are extracted from around 790k news articles in Reuters [9]. We preprocessed the English Reuters articles using the Stanford dependency parser and co-reference resolution [10]. We lemmatized all words, and for subjects and objects we considered only the head words, and ignored words like WH-pronouns.

All relations are defined between two events of the form (S, V_l, O) and (S, V_r, O), where subject S and object O are the same. As candidates we consider only events in sequence (occurrence in text).

Positive Samples We extract positive samples of the form $(S, V_l, O) \rightarrow (S, V_r^{pos}, O)$, if

1. $V_l \rightarrow V_r^{pos}$ is listed in VerbOcean as happens-before relation.
2. $\neg[V_l \Rightarrow V_r^{pos}]$ according to WordNet. That means, for example, if (S, V_r, O) is paraphrasing (S, V_l, O), then this is not considered as a temporal relation.

This way, we were able to extract 1699 positive samples. Examples are shown in Table 2.

Negative Samples Using VerbOcean, we extracted negative samples of the form $(S, V_l, O) \nrightarrow (S, V_r^{neg}, O)$, i.e. the event on the left hand (S, V_l, O) is the same as for a positive sample.[5] This way, we extracted 1177 negative samples.

[5] If $(S, V_l, O) \nrightarrow (S, V_r^{neg}, O)$, then $V_l \rightarrow V_r^{neg}$ is not listed in VerbOcean.

Table 2: Examples of happens-before relations extracted from news articles.

Examples
(company, buy, share) $\rightarrow$ (company, use, share)
(ex-husband, stalk, her) $\rightarrow$ (ex-husband, kill, her)
(farmer, plant, acre) $\rightarrow$ (farmer, harvest, acre)

There are several reasons for a relation not being in a temporal relation. Using VerbOcean and WordNet we analyzed the negative samples, and found that the majority (1030 relations) could not be classified with either VerbOcean or WordNet. We estimated conservatively that around 27% of these relations are false negatives: for a sub-set of 100 relations, we labeled a sample as a false negative, if it can have an interpretation as a happens-before relation.[6]

To simplify the task, we created a balanced data set, by pairing all positive and negative samples: each sample pair contains one positive and one negative sample, and the task is to find that the positive sample is more likely to be a happens-before relation than a negative sample. The resulting data set contains in total 1765 pairs.

4 Analogy-based reasoning for happens-before relation scoring

In the following, let r be a happens-before relation of the form:

$$r : e_l \rightarrow e_r \,,$$

where e_l and e_r are two events of the form (S, V_l, O) and (S, V_r, O), respectively. Furthermore, let e' be any event of the form (S', V', O').

Our working hypotheses consists of the following two claims:

$$\text{(I) If } \; (e' \Rightarrow e_l) \wedge (e_l \rightarrow e_r), \text{ then } \; e' \rightarrow e_r \,.$$
$$\text{(II) If } \; (e' \Rightarrow e_r) \wedge (e_l \rightarrow e_r), \text{ then } \; e_l \rightarrow e' \,.$$

For example, consider

"John buys computer" $\Rightarrow$ "John acquires computer" ,
"John acquires computer" $\rightarrow$ "John uses computer" .

Using (I), we can reason that:

"John buys computer" $\rightarrow$ "John uses computer" .

We note that, in some cases, " $\Rightarrow$" in (I) and (II) cannot be replace by " $\Leftarrow$". This is illustrated by the following example:

"John knows Sara" $\Leftarrow$ "John marries Sara" ,
"John marries Sara" $\rightarrow$ "John divorces from Sara" .

However, the next statement is considered wrong (or less likely to be true):

"John knows Sara" $\rightarrow$ "John divorces from Sara" .

In practice, using word embeddings, it can be difficult to distinguish between " $\Rightarrow$" and " $\Leftarrow$". Therefore, our proposed method uses the following simplified assumptions:

$$\text{(I}^*\text{) If } \; (e' \sim e_l) \wedge (e_l \rightarrow e_r), \text{ then } \; e' \rightarrow e_r \,.$$
$$\text{(II}^*\text{) If } \; (e' \sim e_r) \wedge (e_l \rightarrow e_r), \text{ then } \; e_l \rightarrow e' \,.$$

where $\sim$ denotes some similarity that can be measured by means of word embeddings.

[6] Therefore, this over-estimates the number of false negatives. This is because it also counts a happens-before relation that is less likely than a happens-after relation as a false negative.

4.1 Memory Comparison Network

We propose a memory-based network model that uses the assumptions (I*) and (II*). It bases its decision on one (or more) training samples that are similar to a test sample. In contrast to other methods like neural networks for script learning, and (non-linear) SVM ranking models, it has the advantage of giving an explanation of why a relation is considered (or not considered) as a happens-before relation.

In the following, let r_1 and r_2 be two happens-before relations of the form:

$$r_1 : (S_1, V_{l_1}, O_1) \rightarrow (S_1, V_{r_1}, O_1)\,,$$
$$r_2 : (S_2, V_{l_2}, O_2) \rightarrow (S_2, V_{r_2}, O_2)\,.$$

Let $\mathbf{x}_{s_i}$, $\mathbf{x}_{vli}$, $\mathbf{x}_{v_{r_i}}$ and $\mathbf{x}_{o_i} \in \mathbb{R}^d$ denote the word embeddings corresponding to S_i, V_{l_i}, V_{r_i} and O_i.[7]

We define the similarity between two relations r_1 and r_2 as:

$$sim_{\boldsymbol{\theta}}(r_1, r_2) = g_{\boldsymbol{\theta}}(\mathbf{x}_{v_{l_1}}^T \mathbf{x}_{v_{l_2}}) + g_{\boldsymbol{\theta}}(\mathbf{x}_{v_{r_1}}^T \mathbf{x}_{v_{r_2}})\,, \tag{1}$$

where $g_{\boldsymbol{\theta}}$ is an artificial neuron with $\boldsymbol{\theta} = \{\sigma, \beta\}$, a scale $\sigma \in \mathbb{R}$, and a bias $\beta \in \mathbb{R}$ parameter, followed by a non-linearity. We use as non-linearity the sigmoid function. Furthermore, here we assume that all word embeddings are l2-normalized.

Given the input relation $r : e_l \rightarrow e_r$, we test whether the relation is correct or wrong as follows. Let n_{pos} and n_{neg} denote the number of positive and negative training samples, respectively. First, we compare to all positive and negative training relations in the training data set, and denote the resulting vectors as $\boldsymbol{u}^{pos} \in \mathbb{R}^{n_{pos}}$ and $\boldsymbol{u}^{neg} \in \mathbb{R}^{n_{neg}}$, respectively. That is formally

$$u_t^{pos} = sim_{\boldsymbol{\theta}}(r, r_t^{pos}) \;\text{ and }\; u_t^{neg} = sim_{\boldsymbol{\theta}}(r, r_t^{neg})\,,$$

where r_t^{pos} and r_t^{neg} denotes the t-th positive/negative training sample.

Next, we define the score that r is correct/wrong as the weighted average of the relation similarities:

$$o^{pos} = \text{softmax}_\gamma(\mathbf{u}^{pos})^T \mathbf{u}^{pos} \;\text{ and }\; o^{neg} = \text{softmax}_\gamma(\mathbf{u}^{neg})^T \mathbf{u}^{neg} \tag{2}$$

where $\text{softmax}_\gamma(\mathbf{u})$ returns a column vector with the t-th output defined as

$$\text{softmax}_\gamma(\mathbf{u})_t = \frac{e^{\gamma u_t}}{\sum_i e^{\gamma u_i}}\,,$$

and $\gamma \in \mathbb{R}$ is a weighting parameter. Note that for $\gamma \rightarrow \infty$, $\text{softmax}_\gamma(\mathbf{u}) = \max(\mathbf{u})$, and for $\gamma = 0$, o is the average of $\mathbf{u}$.

Finally, we define the happens-before score for r as

$$l(e_l, e_r) = o^{pos}(e_l, e_r) - o^{neg}(e_l, e_r)\,. \tag{3}$$

The score $l(e_l, e_r)$ can be considered as an unnormalized log probability that relation r is a happens-before relation. The basic components of the network are illustrated in Figure 2.

For optimizing the parameters of our model we minimize the rank margin loss:

$$L(r^{pos}, r^{neg}) = \max\{0, 1 - l(e_l, e_r^{pos}) + l(e_l, e_r^{neg}))\}\,, \tag{4}$$

where $r^{pos} : e_l \rightarrow e_r^{pos}$ and $r^{neg} : e_l \rightarrow e_r^{neg}$ are positive and negative samples from the held-out training data. All parameters of the models are trained using stochastic gradient descent (SGD). Word embeddings ($\mathbf{x}_s, \mathbf{x}_v$, and $\mathbf{x}_o$) are kept fixed during training.

Our model can be interpreted as an instance of the Memory Networks proposed in [17]. Using the notation from [17], $I(\cdot)$ corresponds to the word embedding lookup, $G(\cdot)$ saves all training samples into the memory, the $O(\cdot)$ function corresponds to (o^{pos}, o^{neg}), and the output of $R(\cdot)$ equals Equation (3).

Our model also has similarity to the memory-based reasoning system proposed in [16], with two differences. First, we use here a trainable similarity measure, see Equation (1), rather than a fixed distance measure. Second, we use the trainable softmax_γ rather than max.

[7] Remark about our notation: we use bold fonts, like $\mathbf{v}$ to denote a column vector; $\mathbf{v}^T$ to denote the transpose, and v_t to denote the t-th dimension of $\mathbf{v}$.

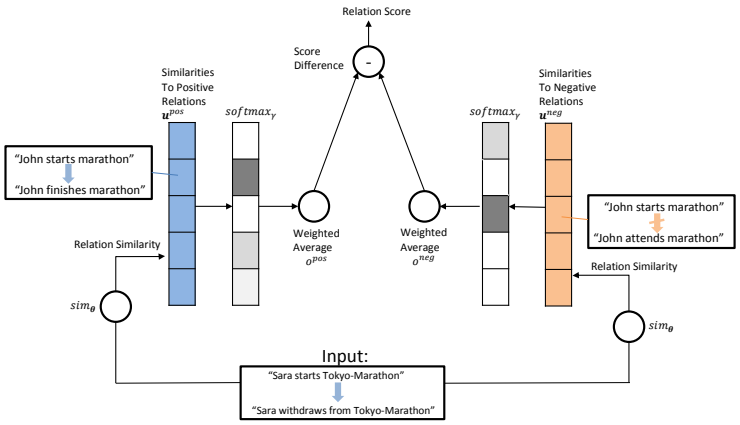

Figure 2: Illustration of proposed model.

5 Alternative ranking models

We also investigate several other models that can be applied for ranking temporal relations. All models that we consider are based on word embeddings in order to be able to generalize to unseen events.

Our first model is based on the bilinear model proposed in [1] for document retrieval, with scoring function $l(e_l, e_r) = \mathbf{z}_l^T M \mathbf{z}_r$, where $\mathbf{z}_l$ and $\mathbf{z}_r$ are the concatenated word embeddings $\mathbf{x}_s, \mathbf{x}_{v_l}, \mathbf{x}_o$ and $\mathbf{x}_s, \mathbf{x}_{v_r}, \mathbf{x}_o$, respectively, and parameter matrix $M \in \mathbb{R}^{3d \times 3d}$. We denote this model as Bai2009.

We also test three neural network architecture that were proposed in different contexts. The model in [2], originally proposed for semantic parsing, is a three layer network that can learn non-linear combinations of (subject, verb) and (verb, object) pairs. The non-linearity is achieved by the Hadamard product of the hidden layers. The original network can handle only events (relations between verb and objects, but not relations between events). We recursively extend the model to handle relations between events. We denote the model as Bordes2012.

In the context of script learning, recently two neural networks have been proposed for detecting happens-before relations. The model proposed in [11] (here denoted Modi2014) learns event embeddings parameterized with verb and context (subject or object) dependent embedding matrices. The event embeddings are then mapped to a score that indicates temporal time. To score a relation between events, we use the dot-product between the two events' embeddings.[8] The model in [6] suggests a deeper architecture than [11]. Their model (denoted here Granroth2016) uses additionally two non-linear layers for combining the left and right events. All neural networks and the Bai2009 model were trained in the same way as the proposed method, i.e. optimized with respect to rank margin loss using Equation (4).[9] For all methods, we kept the word embeddings fixed (i.e. no training), since this improved performance in general.

Our final two models use the rankSVM Algorithm proposed in [7] with the implementation from [8]. We tested both a linear and a rbf-kernel with the hyper-parameters optimized via grid-search. To represent a sample, we concatenate the embeddings of all words in the relation.

6 Experiments

We split the data set into training (around 50%), validation (around 25%), and testing (around 25%) set. Due to the relatively small size of the data we repeated each experiment 10 times for different random splits (training/validation/test).

[8] We also tried two variations: left and right events with different and same parameterization. However, the results did not change significantly.

[9] Originally, the model in [6] was optimized with respect to negative log-likelihood, however in our setting we found that rank-margin loss performed better.

Table 3: Mean accuracy and standard deviation (in brackets) of all methods for 10 random splits of training/validation/test.

Method	**Test Data**	**Validation Data**
Human estimate	76.7%	76.7%
Memory Comparison Network (softmax$_\gamma$, trained)	61.4 (11.5)	75.2 (7.6)
Granroth2016	60.9 (5.3)	72.7 (7.0)
Modi2014[10]	57.5 (7.8)	74.9 (6.6)
Bordes2012	58.3 (7.8)	74.9 (5.5)
Bai2009	58.9 (7.4)	72.7 (7.1)
rankSVM (rbf)	60.8 (6.1)	74.4 (6.7)
rankSVM (linear)	59.1 (9.4)	74.9 (6.5)
Random Baseline	50%	50%
Memory Comparison Network (softmax$_\gamma$, trained)	61.4 (11.5)	75.2 (7.6)
Memory Comparison Network (softmax$_\gamma$, initial parameters)	60.7 (5.9)	69.0 (9.7)
Memory Comparison Network (max, trained)	60.5 (6.2)	67.7 (9.7)
Memory Comparison Network (max, no parameters)	60.1 (5.8)	65.1 (10.9)
Memory Comparison Network (average, no parameters)	60.1 (5.9)	66.5 (8.7)

For the bilinear model and all neural networks, we performed up to 2000 epochs, and used early stopping with respect to the validation set. Some models were quite sensitive to the choice of the learning rates, so we tested 0.00001, 0.0001, and 0.001, and report the best results on the validation set.

For our proposed method, we set the learning rate constant to 0.001. Furthermore, we note that our proposed method requires two types of training data, one type of training data that is in memory, the other type that is used for learning the parameters. For the former and latter we used the training and validation fold, respectively. As initial parameters for this non-convex optimization problem we set $\sigma = 1.0, \beta = -0.5, \gamma = 5.0$, that were selected via the validation set.

For testing, we consider the challenging scenario, where the left event of the sample contains a verb that is not contained in the training set (and also not in the validation set).

We report accuracy, when asking the question: given observation (S, V_l, O), is (S, V_r^{pos}, O) more likely to be a future event than (S, V_r^{neg}, O)?

We used the 50 dimensional word embeddings from GloVe tool [12] trained on Wikipedia + Gigaword 5 provided by the authors (GloVe)[11].

The results of our method and previously proposed methods are shown in Table 3, upper half. By using the false-negative estimate from Section 3.1, we also calculated an estimate of the human performance on this task.[12]

The results suggest that our proposed model provides good generalization performance that is at par with the neural network recently proposed in [6] (Granroth2016), and SVM ranking with RBF-kernel. The results support our claim that the happens-before relation can be detected by analogy-based reasoning.

6.1 Analysis

We also compared to four variations of our proposed method. The results are shown in Table 3, lower half.

The first two variations use as similarity measure the addition of the word embeddings' inner products, i.e. $g_{\boldsymbol{\theta}}$ in Equation (1) is the identity function, and have no trainable parameters. The variation denoted by "Memory Comparison Network (max, no parameters)", is a kind of nearest neighbour ranking, that uses the max function instead of softmax$_\gamma$. The second variation, denoted by "Memory

[11] `http://nlp.stanford.edu/projects/glove/`

[12] We assume that distinguishing a false-negative from a true-positive is not possible (i.e. a human needs to guess), and that all guesses are wrong.

Table 4: Four examples with input relations, output scores and evidences by our proposed method.

input relation: (index,climb,percent) $\rightarrow$ (index,slide,percent)			
o^{pos}**:**	0.745	supporting evidence:	(rate,rise,percent) $\rightarrow$ (rate,tumble,percent)
o^{neg}**:**	0.697	supporting evidence:	(index,finish,point) $\nrightarrow$ (index,slide,point)
input relation: (parliament,discuss,budget) $\rightarrow$ (parliament,adopt,budget)			
o^{pos}**:**	0.412	supporting evidence:	(refiner,introduce,system) $\rightarrow$ (refiner,adopt,system)
o^{neg}**:**	0.352	supporting evidence:	(union,call,strike) $\nrightarrow$ (union,propose,strike)
input relation: (price,gain,cent) $\nrightarrow$ (price,strengthen,cent)			
o^{pos}**:**	0.542	supporting evidence:	(investment,build,plant) $\rightarrow$ (investment,expand,plant)
o^{neg}**:**	0.753	supporting evidence:	(dollar,rise,yen) $\nrightarrow$ (dollar,strengthen,yen)
input relation: (farmer,plant,acre) $\nrightarrow$ (farmer,seed,acre)			
o^{pos}**:**	0.136	supporting evidence:	(refinery,produce,tonne) $\rightarrow$ (refinery,process,tonne)
o^{neg}**:**	0.145	supporting evidence:	(refinery,produce,tonne) $\nrightarrow$ (refinery,receive,tonne)

Comparison Network (average, no parameters)", uses for o^{pos} and o^{neg}, in Equations (2), the average of $\mathbf{u}^{pos}$ and $\mathbf{u}^{neg}$, respectively. The performance of both variations is below our proposed method.

Furthermore, we compared to an alternative model, where the softmax$_\gamma$ is replaced by the max function, marked by "(max, trained)" in Table 3, lower half. Also, we compared to our proposed model, but without learning parameters, i.e. the parameters are set to the initial parameters, marked by "(softmax$_\gamma$, initial parameters)" in Table 3, lower half. We can see that the choice of softmax$_\gamma$, over max, improves performance, and that the training of all parameters with SGD is effective (in particular, see improvement on validation data).

Since our model uses analogy-based reasoning, we can easily identify "supporting evidence" for the output of our system. Four examples are shown in Table 4. Here, "supporting evidence" denotes the training sample with the highest similarity $sim_{\boldsymbol{\theta}}$ to the input. In the first and second example, the input is a happens-before relation, in the third and fourth example, the input is not a happens-before relation.[13]

7 Discussion

Our current method does not model the interaction between subject, object and verb. However, temporal relations can also crucially depend on subject and object. As an example, in our data set (see Table 2), we have the happens-before relation (company, buy, share) $\rightarrow$ (company, use, share). Clearly, if we replace the subject by "kid" and the object by "ice-cream", the happens-before relation becomes wrong, or much less likely. In particular, (kid, buy, ice-cream) $\rightarrow$ (kid, use, ice-cream) is much less likely than, for example, (kid, buy, ice-cream) $\rightarrow$ (kid, eat, ice-cream).[14]

Here, we compared two temporal rules r_1 and r_2 and asked which one is more likely, by ranking them. However, reasoning in terms of probabilities of future events, would allow us to integrate our predictions into a probabilistic reasoning framework like MLN [14].

8 Conclusions

We investigated how common knowledge, provided by lexical resources, can be generalized and used to predict future events. In particular, we proposed a memory network that can learn how to compare and combine the similarity of the input events to event relations saved in memory. This way our proposed method can generalize to unseen events and also provide evidence for its reasoning. Our experiments suggest that our method is competitive to other (deep) neural networks and rankSVM.

[13] Since we considered only the head, a unit like "percent" means "x percent", where x is some number.

[14] Partly, this could be addressed by considering also the selectional preference of verbs like "eat" and "use".

References

[1] Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. Supervised semantic indexing. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 187–196. ACM, 2009.

[2] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*, pages 127–135, 2012.

[3] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797, 2008.

[4] Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *EMNLP*, volume 4, pages 33–40, 2004.

[5] Christiane Fellbaum and George Miller. Wordnet: An electronic lexical database. MIT Press, 1998.

[6] Mark Granroth-Wilding and Clark. What happens next? Event prediction using a compositional neural network model. *AAAI*, 2016.

[7] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[8] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

[9] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[10] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, pages 55–60, 2014.

[11] Ashutosh Modi and Ivan Titov. Inducing neural models of script knowledge. In *CoNLL*, volume 14, pages 49–57, 2014.

[12] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods on Natural Language Processing*, pages 1532–43, 2014.

[13] Karl Pichotta and Raymond J Mooney. Statistical script learning with multi-argument events. In *EACL*, volume 14, pages 220–229, 2014.

[14] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[15] Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. Script induction as language modeling. In *EMNLP*, 2015.

[16] Craig Stanfill and David Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[17] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *ICLR 2015*, 2015.

[18] Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol T. Rutherford. The conll-2015 shared task on shallow discourse parsing. In *CoNLL*, page 2, 2015.