Event participant modelling with neural networks

Ottokar Tilk
Institute of Cybernetics
Tallinn University of Technology
12618 Tallinn, Estonia
ottokar.tilk@phon.ioc.ee

Vera Demberg and **Asad Sayeed**
and **Dietrich Klakow** and **Stefan Thater**
Saarland University
66123 Saarbrücken, Germany
{vera,asayeed,stth}
@coli.uni-sb.de;
dietrich.klakow@lsv.uni-sb.de

Abstract

A common problem in cognitive modelling is lack of access to accurate broad-coverage models of event-level surprisal. As shown in, e.g., Bicknell et al. (2010), event-level knowledge does affect human expectations for verbal arguments. For example, the model should be able to predict that *mechanics* are likely to check *tires*, while *journalists* are more likely to check *typos*. Similarly, we would like to predict what locations are likely for *playing football* or *playing flute* in order to estimate the surprisal of actually-encountered locations. Furthermore, such a model can be used to provide a probability distribution over fillers for a thematic role which is not mentioned in the text at all.

To this end, we train two neural network models (an incremental one and a non-incremental one) on large amounts of automatically role-labelled text. Our models are probabilistic and can handle several roles at once, which also enables them to learn interactions between different role fillers. Evaluation shows a drastic improvement over current state-of-the-art systems on modelling human thematic fit judgements, and we demonstrate via a sentence similarity task that the system learns highly useful embeddings.

1 Introduction

Our goals in this paper are to learn a representation of events and their thematic roles based on large quantities of automatically role-labelled text and to be able to calculate probability distributions over the possible role fillers of specific missing roles. In this sense, the task is closely related to work on selectional preference acquisition (Van de Cruys, 2014). We focus here on the roles *agent*, *patient*, *location*, *time*, *manner* and the *predicate* itself. The model we develop is trained to represent the event-relevant context and hence systematically captures long-range dependencies. This has been previously shown to be beneficial also for more general language modelling tasks (e.g., Chelba and Jelinek, 1998; Tan et al., 2012).

This type of modelling is potentially relevant to a wide range of tasks, for instance for performing thematic fit judgment tasks, detecting anomalous events (Dasigi and Hovy, 2014), or predicting event structure that is not explicitly present in the text. The latter could be useful for inferring missing information in entailment tasks or improving identification of thematic roles outside the sentence containing the predicate. Potential applications also include predicate prediction based on arguments and roles, which has been noted to be relevant for simultaneous machine translation for a verb-final to a verb-medial source language (Grissom II et al., 2014). Within cognitive modelling, our model could help to more accurately estimate *semantic surprisal* for broad-coverage texts, when used in combination with an incremental role labeller (e.g., Konstas and Keller, 2015), or to provide surprisal estimates for content words as a control variable for psycholinguistic experimental materials.

In this work, we focus on the predictability of verbs and nouns, and we suggest that the predictability of these words depends to a large extent on the relationship of these words to other nouns and

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 171–182,
Austin, Texas, November 1-5, 2016.

verbs, especially those connected via the same event. We choose a neural network (NN) model because we found that results from existing related models, e.g. Baroni and Lenci's Distributional Memory, depend heavily on how exactly the distributional space is defined, while having no principled way of optimizing the space. A crucial advantage of a neural network-based approach is thus that the model can be trained to optimize the distributional representation for the task.

Our model is trained specifically to predict missing semantic role-fillers based on the predicate and other available role-fillers of that predicate. The model can also predict the predicate based on the semantic roles and their fillers. In our model, there is no difference in how the semantic roles or the predicate are treated. Thus, when we refer here to *roles*, we usually mean both semantic roles and the predicate, unless otherwise explicitly stated.

Our model is compositional in that it has access to several role-fillers (including the verb) at the same time, and can thus represent interdependencies between participants of an event and predict from a combined representation. Consider, for example, the predicate *serve*, whose likely patients include e.g., *drinks*. If we had the agent *robber*, we would like to be able to predict a patient like *sentence*, in the sense of "the robber will serve his sentence..." This task is related to modelling thematic fit. In this paper, we evaluate our model on a variety of thematic fit rating datasets as well as on a sentence similarity dataset that tests for successful compositionality in our model's representations.

This paper makes the following contributions:

- We compare two novel NN models for generating a probability distribution over selectional preferences given one or more roles and fillers.
- We show that our technique outperforms state of the art thematic fit models on many datasets.
- We show that the embeddings thus obtained are effective in measuring sentence similarity.

1.1 Neural networks

Neural networks have proven themselves to be very well suited for language modeling. By learning distributed representations of words (Bengio et al., 2003), they are able to generalize to new contexts that were not observed word-by-word in the training corpus. They can also use a relatively large number of context words in order to make predictions about the upcoming word. In fact, the recurrent neural network (RNN) LM (Mikolov et al., 2010) does not explicitly fix the context size at all but is potentially able to compress the relevant information about the entire context in its recurrent layer. These are the properties that we would like to see in our role-filler prediction model as well.

Neural networks have also been used for selectional preference acquisition, as in Van de Cruys (2014). His selectional preference model differs from our model in several aspects. First, unlike our model it is limited to a fixed number of inputs. Another difference is that his model uses separate embeddings for all input words, while ours enables partial parameter sharing. Finally and crucially for role-filler prediction, selectional preference models score the inputs, while our model gives a probability distribution over all words for the queried target role.

We discuss the components necessary for our model in more detail in section 3.

2 Data source

Our source of training data is the ukWaC corpus, which is part of the WaCky project, as well as the British National Corpus. The corpus consists of web pages crawled from the `.uk` web domain, containing approximately 138 million sentences.

These sentences were run through a semantic role labeller and head words were extracted as described in Sayeed et al. (2015). The semantic role labeller used, SENNA (Collobert and Weston, 2007), generates PropBank-style role labels. While PropBank argument positions (ARG0, ARG1, etc.) are primarily designed to be verb-specific, rather than directly representing "classical" thematic roles (agent, patient, etc.), in the majority of cases, ARG0 lines up with agent roles and ARG1 lines up with patient roles. PropBank-style roles have been used in other recent efforts in thematic fit modelling (e.g., Baroni et al., 2014; Vandekerckhove et al., 2009),

For processing purposes, the corpus was divided into 3500 segments. Fourteen segments (approx 500 thousand sentences) each were used for development and testing, and the rest were used for training.

In order to construct our incremental model and compare it to n-gram language models, we needed a precise mapping between the lemmatized argument words and their positions in the original sentence. This required aligning the SENNA tokenization and the original ukWaC tokenization used for MaltParser. Because of the heterogeneous nature of web data, this alignment was not always achievable—we skipped a small number of sentences in this case. In the development and testing portions of the data set, we filtered sentences containing predicates where there were multiple role-assignees with the same role for the same predicate.

3 Model design and implementation

Our model is a neural network with a single nonlinear hidden layer and a *Softmax* output layer. All inputs are one-hot encoded—i.e., represented as a binary vector with size equal to the number of possible input values, where all entries are zero except the entry at the index corresponding to the current input value.

3.1 Two-part view of the model

The parameters of a neural network classifier with a single hidden layer and one-hot encoded inputs can be viewed as serving two distinct purposes: moving from inputs towards outputs, the first weight matrix that we encounter is responsible for learning distributed representations (or embeddings) of the inputs; the second weight matrix represents the parameters of a maximum entropy classifier that uses the learned embeddings as inputs.

Considering the task of role-filler prediction, we would want these two sets of parameters to have the following properties:

- The **classifier layer** should be different for each target role, because the suitable filler given the context can clearly be very different depending on the role (e.g., verb vs. agent).
- The **embedding layer** should also be different depending on the role of context word. Otherwise, the network would not have any information about the role of the context word. For example, the suitable verb filler for context word *dog* in an agent role is probably very different from what it would be, were it in a patient role (e.g. *bark* vs. *feed*).

We now briefly describe some incrementally improved intermediate approaches that we also considered as they help to understand the steps that led to our final solution for achieving the desired properties of the embedding and classifier layer.

A naive way to accomplish the aspired properties would be to have a separate model for each input role and target role pair. This approach has several drawbacks. For a start, there is no obvious way to model interactions of different input roles and fillers in order to make predictions based on multiple input role-word pairs simultaneously. Another problem is that the parameters are trained only on a fraction of available training data—e.g., verb embedding weights are trained independently for each target role classifier. Finally, given that we have chosen to distinguish between n different roles, it would require us to train and tune hyper-parameters for n^2 models.

One of these problems (data under-utilization) can be alleviated by sharing role-specific embedding and classifier weights across different models. For example, the verb embedding matrix would be shared across all models that predict different role fillers based on input verbs. Other problems remain, and training the large number of models becomes even more difficult because of parameter synchronization, but this is a step towards the next improvement.

Shared role-specific embedding and classifier weights enable us to combine all input-target role pair models into a single model. This can be done by stacking role-specific embedding matrices to form a 3-way embedding tensor and building a classifier parameter tensor analogously. Having a single model saves us from tuning multiple models and makes modelling interactions between inputs possible.

Despite these advantages, having two tensors in our model has a drawback of rapidly growing the number of parameters as vocabulary size, number of roles, and hidden layer size increase. This may lead to over-fitting and increases training time.

A more subtle weakness is the fact that this kind of model lacks parameter sharing across role-specific embedding weight matrices. It is clear that some characteristics of words (e.g., semantics) usu-

ally remain the same across different roles. Thus it is practical to share some information across role-specific weights so that the embeddings can benefit from more data and learn better semantic representations while leaving room for role-specific traits.

For these reasons we replace the tensors with their factored form in our models.

3.2 Factored parameter tensors

Factoring classifier and embedding tensors helps to alleviate both the efficiency and parameter sharing problems brought out in Section 3.1.

Given vocabulary size $|V|$, number of roles $|R|$ and hidden layer size H, each tensor T would require $|V| \times |R| \times H$ parameters. The number of parameters can be reduced by expressing the tensor as a sum of F rank-one tensors (Hitchcock, 1927). This technique enables us to replace the tensor T with three factor matrices A, B and C. Each tensor element $T[i, j, k]$ can then be written as:

$$T[i, j, k] = \sum_{f=1}^{F} A[i, f]B[j, f]C[f, k] \quad (1)$$

Assuming lateral slices of T represent role-specific weight matrices (index j denotes roles), we write each role specific weight matrix W as:

$$W = A \,\mathrm{diag}(rB)C \quad (2)$$

where r is a one-hot encoded role vector and $diag$ is a function that returns a square matrix with the argument vector on the main diagonal and zeros elsewhere. For example, with a vocabulary of 50000 words, 7 roles and number of factors and hidden units equal to 512, the factorization reduces the number of parameters from 179M to 26M and greatly improves training speed. Factorization also enables parameter sharing, since factor matrices A and C are shared across all roles.

Factored tensors have been used in different neural network models before. Starting with restricted Boltzmann machines, Memisevic and Hinton (2010) used a factored 3-way interaction tensor in their image transformation model. Sutskever et al. (2011) created a character level RNN LM that was efficiently able to use input character specific recurrent weights by using a factored tensor. Alumäe (2013) used a factored tensor in a multi-domain LM to be able to use a domain-specific hidden layer weight matrix that would take into account the differences while exploiting similarities between domains. A multi-modal LM by Kiros et al. (2014) uses a factored tensor to change the effective output layer weights based on image features.

It has been noticed before, that training models with factored tensors as parameters using gradient descent is difficult (Sutskever et al., 2011; Kiros et al., 2014). As explained by Sutskever et al. (2011), this is caused by the fact that each tensor element is represented as a product of three parameters, which may cause disproportionate updates if these three factors have magnitudes that are too different. Another problem is that if the factor matrix B happens to have too small or too large values, then this might also cause instabilities in the lower layers as the back-propagated gradients are scaled by role-specific row of B in our model. This situation is magnified in our models, since we have not one, but two factored layers.

To solve this problem, Sutskever et al. (2011) suggest using 2nd order methods instead of gradient descent. Alumäe (2013) has alleviated the problem of shrinking back-propagated gradients by adding a bias (initialized with ones) to the domain-specific factor vector. We found that using AdaGrad (Duchi et al., 2011) to update the parameters is very effective. The method provides parameter-specific learning rates that depend on the historic magnitudes of the gradients of these parameters. This seems to neutralize the effect of vanishing or exploding gradients by reducing the step size for parameters that tend to have large gradients and allow a bigger learning rate for parameters with smaller gradients.

3.3 General structure of the model

Our general approach, common to both role-filler models, is shown in Figure 1. First, role-specific word embedding vector e is computed by implicitly taking a fiber (word indexed row of a role indexed slice) from the factored embedding tensor:

$$e = wA_e \,\mathrm{diag}(rB_e)C_e \quad (3)$$

$$h = \mathrm{PReLU}(e + b_h) \quad (4)$$

where w and r are one-hot encoded word and role vectors respectively, b_h is hidden layer bias, and A_e,

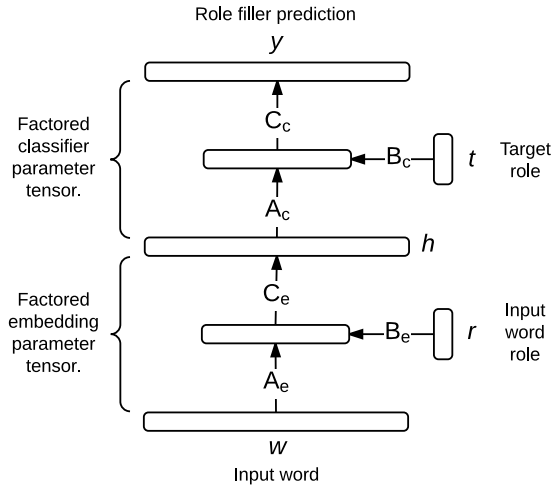

Figure 1: General structure of role-filler models.

B_e and C_e represent the factor matrices that the embedding tensor is factored into. Next, we apply a parametric rectifier (PReLU; He et al., 2015) nonlinearity to the role-specific word embedding to obtain the hidden activation vector h.

The hidden layer activation vector h is fed to the *Softmax* output layer through a target role specific classifier weight matrix (a target role-indexed slice of the classifier parameter tensor):

$$c = hA_c \,\text{diag}(tB_c)C_c \quad (5)$$

$$y = \text{Softmax}(c + b_y) \quad (6)$$

where t is a one-hot encoded target role vector, b_y is output layer bias, and y is the output of the model representing the probability distribution over the output vocabulary.

3.4 Modeling input interactions

The general approach described in Section 3.3 also allows us to model interactions between different input role-word pairs. If we know the order in which the inputs were introduced, then we can add a recurrent connection to the hidden layer to implement an incremental role filler predictor. When word order is unknown, then input role-word pair representations can be added together to compose the representation of the entire predicate context[1]. We chose addition over concatenation (often preferred in language models) because the non-incremental model does not need to preserve information about word order, and addition also enables using a variable number of inputs.

The incremental model adds information about the previous hidden state h_{t-1} to the current input word role-specific embedding e_t through recurrent weights W_r. So, Equation 4 is replaced with:

$$h_t = \text{PReLU}(e_t + h_{t-1}W_r + p_tW_p + b_h) \quad (7)$$

where p_t is a binary predicate boundary indicator that informs the model about the start of a new predicate and equals 1 when the target word belongs to a new predicate and 0 otherwise. The predicate boundary input p_t is connected to the network through parameter vector W_p. The hidden state h_0 is initialized to zeros.

The non-incremental model adds role-specific embedding vectors of all input words together to form the representation of the entire predicate context and replaces Equation 4 with:

$$h = \text{PReLU}(\sum_{i=1}^{N} e_i + b_h) \quad (8)$$

where N is the number of input role-word pairs.

3.5 Training details

First, we give details that are common to both the RNN and NN models. The models are trained with mini-batches of 128 samples. The hidden layer consists of 256 PReLU units; embedding and classifier tensor factorization layer sizes are 256 and 512 respectively. The input and output vocabularies are the same, consisting of 50,000 most frequent lemmatized words in the training corpus. The role vocabulary consists of 5 argument roles (ARG0, ARG1, ARGM-LOC, ARGM-TMP and ARGM-MNR), the verb is treated as the sixth role, and all the other roles are mapped to a shared OTHER label. Parameters are updated using AdaGrad (Duchi et al., 2011) with a learning rate of 0.1. All models are implemented using Theano (Bastien et al., 2012; Bergstra et al., 2010) and trained on GPUs for 8 days.

RNN model gradients are computed using backpropagation through time (Rumelhart et al., 1986)

[1] In applications like natural language generation, for example, where role-fillers need to be predicted, it is not necessarily always the case that the order will be known in advance or that the thematic fit model will be used to generate the full sentence in correct word order.

Model Name	Dev	Test
3-gram LM	450.1 ± 2.6	438.9 ± 2.6
3-gram CWM	859.6 ± 4.6	834.9 ± 4.5
RNN CWM	485.8 ± 2.7	473.2 ± 2.6
RNN RF	244.6 ± 1.4	237.8 ± 1.4
NN RF	248.2 ± 1.4	241.9 ± 1.4

Table 1: Perplexities on dev/test dataset.

over 3 time steps. The NN model is trained on mini-batches of 128 samples that are randomly drawn with replacement from the training set.

3.6 Model comparison

Perplexity allows us to compare all our models in similar terms, and evaluate the extent to which access to thematic roles helps the model to predict missing role fillers. For comparability, the perplexities of all models are computed only on content word probabilities (i.e., predicates and their arguments). We also report the 95% confidence interval for perplexity, which is computed according to Klakow and Peters (2002). All models are trained on exactly the same sentences of lemmatized words. Probability mass is distributed across the vocabulary of the 50,000 most frequent content words in the training corpus.

3.6.1 Models

First, we compare our model to a conventional 3-gram language model **3-gram LM**, conditioning on the previous context containing the immediately preceding context of content and function words. All n-grams are discounted with Kneser-Ney smoothing, and n-gram probability estimates are interpolated with lower order estimates. Sentence onset in all models is padded with a special sentence onset tag. The vocabulary of context words for this model consists of all words from the training corpus.

As a second model, we train a 3-gram content word model **3-gram CWM**, which is an N-gram LM that is trained only on content words.

Next, we have **RNN CWM**—an RNN LM (Mikolov et al., 2010) trained on content words only. The context size of this model is not explicitly defined and the model can potentially utilize more context words than 3-gram CWM (even from outside the sentence boundary).

Our incremental role-filler **RNN RF** is similar to RNN CWM, except for using role-specific embedding and classifier weights (slices of factored tensor). It thus has additional information about the content word roles[2].

Finally, the non-incremental role-filler **NN RF** loses the information about word order and the ability to use information outside predicate boundaries and trades it for the ability to see the future (i.e., the context includes both the preceding and the following content words and their roles).

3.6.2 Results

The results of content word perplexity evaluation are summarized in Table 1. The thematic-role informed models outperform all other models by a very large margin, cutting perplexity almost in half. The incremental model achieves a slightly lower perplexity than the non-incremental one (237.8 vs. 241.9), hinting that the content word order and out-of-predicate role-word pairs can be even more informative than a preview of upcoming role-word pairs.

The difference between normal LM and the CWM can be explained by the loss of information from function words, combined with additional sparsity in the model because content word sequences are much sparser than sequences of content and function words.

This also explains why using a neural network-based RNN CWM model improves the performance so much (perplexity drops from 834.9 to 473.2), as neural network based language models are well known for their ability to generalize well to unseen contexts by learning distributed representations of words (Bengio et al., 2003).

4 Evaluation on thematic fit ratings

In order to see whether our model accurately represents events and their typical thematic role fillers, we evaluate our model on a range of existing datasets containing human thematic fit ratings. This evaluation also allows us to compare our model to existing models that have been used on this task.

[2] A reviewer kindly points out, as a matter of historical interest, that the high-level architecture of the RNN RF model bears some resemblance to the parallel distributed processing model in McClelland et al. (1989) and St. John and McClelland (1990).

Data source	# ratings	Roles	NN RF	BL2010	GSD2015	BDK2014
Pado (agent, patient)	414	ARG0, ARG1, ARG2	0.52 (8)	**0.53** (0)	**0.53** (0)	0.41
McRae (agent, patient)	1444	ARG0, ARG1	**0.38** (20)	0.32 (70)	0.36 (70)	0.28
Ferretti (location)	274	ARGM-LOC	**0.44** (3)	0.23 (3)	0.29 (3)	-
Ferretti (instrument)	248	ARGM-MNR	**0.45** (6)	0.36 (17)	0.42 (17)	-
Greenberg (patient)	720	ARG1	**0.61** (8)	0.46 (18)	0.48 (18)	-
Pado+McRae+Ferretti	2380		**0.41** (37)	0.35 (90)	0.38 (90)	-

Table 2: Thematic fit evaluation scores, consisting of Spearman's ρ correlations between average human judgements and model output, with numbers of missing values (due to missing vocabulary entries) in brackets. The baseline scores come from the TypeDM (Baroni and Lenci, 2010) model, further developed and evaluated in Greenberg et al. (2015a,b) and the neural network *predict* model described in Baroni et al. (2014). NN RF is the non-incremental model presented in this article. Our model maps ARG2 in Pado to OTHER role. Significances were calculated using paired two-tailed significance tests for correlations (Steiger, 1980). NN RF was significantly better than both of the other models on the Greenberg and Ferretti location datasets and significantly better than BL2010 but not GSD2015 on McRae and Pado+McRae+Ferretti; differences were not statistically significant for Pado and Ferretti instruments.

4.1 Related work

State-of-the-art computational models of thematic fit quantify the similarity between a role filler of a verb and the proto-typical filler for that role for the verb based on distributional vector space models. For example, the thematic fit of *grass* as a patient for the verb *eat* would be determined by the cosine of a distributional vector representation of *grass* and a prototypical patient of *eat*. The proto-typical patient is in turn obtained from averaging representations of words that typically occur as a patient of *eat* (e.g., Erk, 2007; Baroni and Lenci, 2010; Sayeed and Demberg, 2014; Greenberg et al., 2015b). For more than one role, information from both the agent and the predicate can be used to jointly to predict a patient (e.g., Lenci, 2011).

4.2 Data

Previous studies obtained thematic fit ratings from humans by asking experimental participants to rate how common, plausible, typical, or appropriate some test role-fillers are for given verbs on a scale from 1 (least plausible) to 7 (most plausible) (McRae et al., 1998; Ferretti et al., 2001; Binder et al., 2001; Padó, 2007; Padó et al., 2009; Vandekerckhove et al., 2009; Greenberg et al., 2015a). The datasets include agent, patient, location and instrument roles. For example, in the Padó et al. (2009) dataset, the noun *sound* has a very low rating of 1.1 as the subject of *hear* and a very high rating of 6.8 as the object of *hear*. Each of the verb-role-noun triples was rated by several humans, and our evaluations are done against the average human score. The datasets differ from one another in size (as shown in Table 2), choice of verb-noun pairs, and in how exactly the question was asked of human raters.

4.3 Methods

A major difference between what the state-of-the-art models do and what our model does is that our model distributes a probability mass of one across the vocabulary, while the thematic fit models have no such overall constraint; they will assign a high number to all words that are similar to the prototypical vector, without having to distribute probability mass. Specifically, this implies that two synonymous fillers, one of which is a frequent word like *fire*, and the other of which is an infrequent word, e.g., *blaze*, will get similar ratings by the distributional similarity models, but quite different ratings by the neural network model, as the more frequent word will have higher probability. Greenberg et al. (2015a) showed that human ratings are insensitive to noun frequency. Hence, we report results that adjust for frequency effects by setting the output layer bias of the neural network model to zero. Since the output unit biases of the neural network model are independent from the inputs, they correlate strongly ($r_s = 0.74, p = 0.0$) with training corpus word frequencies after being trained. Therefore, setting the learned output layer bias vector to a zero-vector is a simple way to reduce the effect of word frequencies on the model's output probability distribution.

Role	# ratings	ρ (# NaN)
ARG0	924	0.38 (14)
ARG1	1615	0.51 (22)
ARG2	39	0.59 (0)
ARGM-MNR	248	0.45 (6)
ARGM-LOC	274	0.44 (3)
ALL	3100	0.45 (45)

Table 3: Per role thematic-fit evaluation scores in terms of Spearmans ρ correlations between average human judgements and model output.

4.4 Results

We can see that the neural network model outperforms the baselines on all the datasets except the Pado dataset. An error analysis on the role filler probabilities generated by the neural net points to the effect of level of constraint of the verb on the estimates. For a relatively non-constraining verb, the neural net model will have to distribute the probability mass across many different suitable fillers, while the semantic similarity models do not suffer from this. This implies that filler fit is not directly comparable across verbs in the NN model (only filler predictability is comparable).

Per role results are shown in Table 3. Surprisingly, the model output has the highest correlation with the averaged human judgements for the target role ARG2, despite the fact that ARG2 is mapped to OTHER along with several other roles. The model struggles the most when it comes to predicting fillers for ARG0. There is no noticeable correlation between the role-specific performance and the role occurrence frequency in the samples of our training set. This implies that parameter sharing between roles does indeed help when it comes to balancing the performance between rare and ubiquitous roles as discussed in section 3.1.

4.5 Compositionality

The above thematic role fit data sets only assess the fit between two words. Our model can however also model the interaction between different roles; see Figure 2 for an example of model predictions. We are only aware of one small dataset that can be used to systematically test the effectiveness of the compositionality for this task. The Bicknell et al. (2010) dataset contains triples like *journalist check spelling* vs. *mechanic check spelling* and *journalist check tires* vs. *mechanic check tires* together with human congruity judgments.

Model	NN RF	Lenci 2011
Accuracy 1	**0.687**	0.671
Accuracy 2	0.828	**0.844**

Table 4: Accuracies on the Bicknell evaluation task.

The goal in this task is for the model to reproduce the human judgments on the 64 sentence pairs. Lenci (2011), which we compare against in Table 4, proposed a first compositional model based on TypeDM to evaluate on this task.

We use two accuracy scores for the evaluation, which we call "Accuracy 1" and "Accuracy 2". "Accuracy 1" counts a hit iff the model assigns the composed subject-verb combination a higher score when we test a human-rated better-fitting object in contrast with when we test a worse-fitting one; in other words, a hit is achieved when *journalist check spelling* should be better than *journalist check tires*, if we give the model *journalist check* as the predicate to test against different objects. (The result from Lenci for this task was transmitted by private communication.)

"Accuracy 2" counts a hit iff, given an object, the composed subject-verb combination gives a higher score when the subject is better fitting. That is, a hit is achieved when *journalist check spelling* has a higher score than *mechanic check spelling*, setting the query to the model as *journalist check* and *mechanic check* and finding a score for *spelling* in that context. This accuracy metric is proposed and evaluated in Lenci (2011).

Evaluation shows that our model performs similarly to that of Lenci, although only limited conclusions can be drawn due to the small data set size.

5 Evaluation of event representations: sentence similarity

To show that our model learns to represent input words and their roles in a useful way that reflects the meaning and interactions between inputs, we evaluate our non-incremental model on a sentence similarity task from Grefenstette and Sadrzadeh (2015).

We assign similarity scores to sentence pairs by computing representations for each sentence by tak-

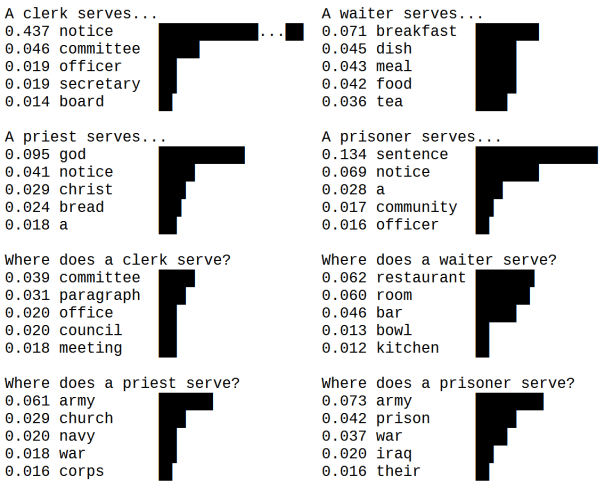

Figure 2: Examples of model predictions for the verb *serve* with different agents and target roles *patient* and *location*.

ing the hidden layer state (Equation 8) of the non-incremental model given the words in the sentence and their corresponding roles. Sentence similarity is then rated with the cosine similarity between the representations of the two sentences.

Spearman's rank correlation between the cosine similarities produced by our model and human ratings are shown in Table 5. Our model achieves much higher correlation with human ratings than the best result reported by Grefenstette and Sadrzadeh (2015), showing our model's ability to compose meaningful representations of multiple input words and their roles.

We also compare our model with another NN word representation model baseline that does not embed role information; by this comparison, we can determine the size of the improvement brought by our role-specific embeddings. The baseline sentence representations are constructed by element-wise addition of pre-trained word2vec (Mikolov et al., 2013) word embeddings[3]. Scores are again computed by using cosine similarity. The large gap between our model's and word2vec baseline's performance illustrates the importance of embedding role information in word representations.

6 Conclusions

In this paper we proposed two neural network architectures for learning proto-typical event representations. These models were trained to generate probability distributions over role fillers for a given semantic role. In our perplexity evaluation, we demonstrated that giving the model access to thematic role information substantially improved prediction performance. We also compared the performance of our model to the performance of current state-of-the-art models in predicting human thematic fit ratings and showed that our model outperforms the existing models by a large margin. Finally, we also showed that the event representations from the hidden layer of our model are highly effective in a sentence similarity task. In future work, we intend to test the potential contribution of this model when applied to larger tasks such as entailment and inference tasks as well as semantic surprisal-based prediction tasks.

[3] https://code.google.com/p/word2vec/

# ratings	NN RF	Kronecker	W2V	Humans
199	**0.34**	0.26	0.13	0.62

Table 5: Sentence similarity evaluation scores on GS2013 dataset (Grefenstette and Sadrzadeh, 2015), consisting of Spearman's ρ correlations between human judgements and model output. Kronecker is the best performing model from Grefenstette and Sadrzadeh (2015). NN RF is the non-incremental model presented in this article, and W2V is the word2vec baseline. Human performance (inter-annotator agreement) shows the upper bound.

7 Acknowledgements

This research was funded by the German Research Foundation (DFG) as part of SFB 1102: "Information Density and Linguistic Encoding" as well as the Cluster of Excellence "Multimodal Computing and Interaction" (MMCI). Also, the authors wish to thank the anonymous reviewers whose valuable ideas contributed to this paper.

References

Alumäe, T. (2013). Multi-domain neural network language model. In *INTERSPEECH*, pages 2182–2186. Citeseer.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 238–247.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation.

Bicknell, K., Elman, J. L., Hare, M., McRae, K., and Kutas, M. (2010). Effects of event knowledge in processing verbal arguments. *Journal of Memory and Language*, 63(4):489–505.

Binder, K. S., Duffy, S. A., and Rayner, K. (2001). The effects of thematic fit and discourse context on syntactic ambiguity resolution. *Journal of Memory and Language*, 44(2):297–324.

Chelba, C. and Jelinek, F. (1998). Exploiting syntactic structure for language modeling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 225–231. Association for Computational Linguistics.

Collobert, R. and Weston, J. (2007). Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 560–567, Prague, Czech Republic. Association for Computational Linguistics.

Dasigi, P. and Hovy, E. H. (2014). Modeling newswire events using neural networks for anomaly detection. In *COLING*, pages 1414–1422.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 216–223, Prague, Czech Republic. Association for Computational Linguistics.

Ferretti, T. R., McRae, K., and Hatherell, A. (2001). Integrating verbs, situation schemas, and thematic role concepts. *Journal of Memory and Language*, 44(4):516–547.

Greenberg, C., Demberg, V., and Sayeed, A. (2015a). Verb polysemy and frequency effects in thematic fit modeling. In *Proceedings of the 6th Workshop on Cognitive Modeling and Computational Linguistics*, pages 48–57, Denver, Colorado. Association for Computational Linguistics.

Greenberg, C., Sayeed, A., and Demberg, V. (2015b). Improving unsupervised vector-space thematic fit evaluation via role-filler prototype clustering. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT)*.

Grefenstette, E. and Sadrzadeh, M. (2015). Concrete models and empirical evaluations for the categorical compositional distributional model of meaning. *Computational Linguistics*.

Grissom II, A. C., Boyd-Graber, J., He, H., Morgan, J., and Daumé III, H. (2014). Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1352.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*.

Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, (6):164–189.

Kiros, R., Salakhutdinov, R., and Zemel, R. (2014).

Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603.

Klakow, D. and Peters, J. (2002). Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19–28.

Konstas, I. and Keller, F. (2015). Semantic role labeling improves incremental parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1191–1201, Beijing, China. Association for Computational Linguistics.

Lenci, A. (2011). Composing and updating verb argument expectations: A distributional semantic model. In *Proceedings of the 2Nd Workshop on Cognitive Modeling and Computational Linguistics*, CMCL '11, pages 58–66, Stroudsburg, PA, USA. Association for Computational Linguistics.

McClelland, J. L., St. John, M., and Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and cognitive processes*, 4(3-4):SI287–SI335.

McRae, K., Spivey-Knowlton, M. J., and Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38(3):283–312.

Memisevic, R. and Hinton, G. E. (2010). Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 22(6):1473–1492.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Padó, U. (2007). *The integration of syntax and semantic plausibility in a wide-coverage model of human sentence processing*. PhD thesis, Saarland University.

Padó, U., Crocker, M. W., and Keller, F. (2009). A probabilistic model of semantic plausibility in sentence processing. *Cognitive Science*, 33(5):794–838.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *NATURE*, 323:9.

Sayeed, A. and Demberg, V. (2014). Combining unsupervised syntactic and semantic models of thematic fit. In *Proceedings of the first Italian Conference on Computational Linguistics (CLiC-it 2014)*.

Sayeed, A., Demberg, V., and Shkadzko, P. (2015). An exploration of semantic features in an unsupervised thematic fit evaluation framework. In *IJCoL vol. 1, n. 1 december 2015: Emerging Topics at the First Italian Conference on Computational Linguistics*, pages 25–40. Accademia University Press.

St. John, M. F. and McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, 46(1-2):217–257.

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87(2):245.

Sutskever, I., Martens, J., and Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Tan, M., Zhou, W., Zheng, L., and Wang, S. (2012). A scalable distributed syntactic, semantic, and lexical language model. *Computational Linguistics*, 38(3):631–671.

Van de Cruys, T. (2014). A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35.

Vandekerckhove, B., Sandra, D., and Daelemans, W. (2009). A robust and extensible exemplar-based model of thematic fit. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 826–834.