View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Apollo

Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

What Happens Next? Event Prediction Using a Compositional Neural Network Model

Mark Granroth-Wilding and **Stephen Clark**
{mark.granroth-wilding, stephen.clark}@cl.cam.ac.uk
Computer Laboratory, University of Cambridge, UK

Abstract

We address the problem of automatically acquiring knowledge of event sequences from text, with the aim of providing a predictive model for use in narrative generation systems. We present a neural network model that simultaneously learns embeddings for words describing events, a function to compose the embeddings into a representation of the event, and a coherence function to predict the strength of association between two events.

We introduce a new development of the narrative cloze evaluation task, better suited to a setting where rich information about events is available. We compare models that learn vector-space representations of the events denoted by verbs in chains centering on a single protagonist. We find that recent work on learning vector-space embeddings to capture word meaning can be effectively applied to this task, including simple incorporation of a verb's arguments in the representation by vector addition. These representations provide a good initialization for learning the richer, compositional model of events with a neural network, vastly outperforming a number of baselines and competitive alternatives.

Introduction

This paper follows a line of work begun by Chambers and Jurafsky (2008), who introduced a technique for automatically extracting knowledge about typical sequences of events from text. They observed that coreference resolution, which identifies passages that make repeated mentions of the same entity, provides a source of information about where an entity participates in multiple events within a document. They make an assumption that places where an entity is mentioned as the argument to a verb denote events that the entity is involved in. In this way, they extract chains of events, each centered on a single entity. They use this data to infer prototypical narrative chains that are similar to the classical notion of a *script* (Schank and Abelson 1977).

A narrative generation system builds the structure of a story at some level of abstraction, involving sequences of events, actions and states, and the characters and entities that participate in them. Planning-based approaches (Turner 1994; Pérez y Pérez and Sharples 2001; Gervás et al. 2005) require knowledge of actions and events that are possible at particular points in the narrative and their effects on the state. Ensuring coherence in generated sequences of events is crucial (Pérez y Pérez and Sharples 2004). Knowledge representations are typically built by hand for small domains, but acquiring them automatically from text is a difficult task. Unsupervised induction in the style of Chambers and Jurafsky (2008) can provide a useful approximation to one part of this knowledge (McIntyre and Lapata 2009).

Ultimately, we aim to provide tools for broad-domain narrative generation. We consider a component that takes a context of events/actions concerning a character (the story so far) and measures the plausibility of possible continuations. This is related to the *narrative cloze* task (Chambers and Jurafsky 2008). In particular, we are interested in the first half of the schema extraction pipeline, which addresses the narrative cloze prediction task, but less in the subsequent step, which infers generalized script-like schemas.

This paper makes two key contributions. Firstly, we propose a new task, *multiple choice narrative cloze* (MCNC), closely related to the narrative cloze task, but better suited to comparing systems' usefulness as a component in narrative generation. In MCNC, a system is able to make use of richer information about events (in both the context and predictions). Secondly, we present a neural network model for predicting whether or not two events are expected to appear in the same chain by learning vector representations of event predicates and argument nouns and a composition function that builds a dense vector representation of the event.

We evaluate a range of systems that induce vector-space representations of events and use them to make predictions, comparing the results to the positive pointwise mutual information (PPMI) measure of Chambers and Jurafsky (2008, henceforth C&J08). We find that the vector-learning system of Mikolov et al. (2013a), `word2vec`, learns useful representations of events for the MCNC task. We show that a naive use of the same system to learn representations of verb arguments yields an improvement in prediction accuracy. We then treat these representations as a starting point for the more sophisticated neural network model that composes information about an event from a verb and its arguments into a single vector. The composed representation can be used for measuring the plausibility of possible next events, vastly outperforming the other models.

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Text: Robbers made a big score, fleeing after stealing more than $300,000 from Wells Fargo armored-truck guards who were servicing the track's ATMs, the Police Department said. The two Wells Fargo guards reported they were starting to put money in the clubhouse ATM when a man with a gun approached and ordered them to lie down. . .

Entity mentions: {*Wells Fargo armored-truck guards*, *The two Wells Fargo guards*, *they*, . . .}

Predicate events: service(x_0, ATMs), report(x_0), put(x_0, money, in clubhouse), lie+down(x_0), . . .

C&J08 events: *(service, subj)*, *(report, subj)*, *(put, subj)*, *(lie+down, subj)*

Figure 1: Example event chain extraction. Mentions of an entity x_0 (dashed underline) are identified by coreference resolution. Events are extracted from verbs to which the entity is an argument (solid underline). The chain is shown first as predicates with arguments, then as predicate-GRs as used by C&J08.

Related Work

Recent work on unsupervised inference of prototypical event sequence information from text began with Chambers and Jurafsky (2008). They introduced a technique for inducing models of event sequences, or *chains*, using coreference resolution, and an evaluation task they call *narrative cloze*. Chambers and Jurafsky (2009) built on this to bring the induced representations closer to the idea of semantic frames, inferring *event schemas*. We focus on the original work of Chambers and Jurafsky (2008), as we are interested in comparing or predicting possible upcoming events in a specific narrative context, rather than building the abstract representation of an event schema. Chambers (2013) and Cheung, Poon, and Vanderwende (2013) also focus on schema induction, rather than next event prediction, but we have explored related models in the present context.

A variety of developments of C&J08 have been proposed. Jans et al. (2012) compare ways of collecting and using the model's statistics to measure association between events. They achieve better results with a bigram conditional probability model than C&J08's PPMI statistic, and we adopt this as one of our baselines.

Balasubramanian et al. (2013) use open-domain relations extracted by the information extraction system Ollie instead of verbs, capturing more information about events. They too focus on event schema extraction. Pichotta and Mooney (2014) propose a solution to limitations of C&J08's representations closer to the original model, estimating the joint probability of a pair of events, taking into account all the entities that they share as arguments (rather than just one, as in C&J08).

Another line of work approaches event knowledge acquisition using *event schema descriptions* (ESDs), natural language descriptions of typical sequences of events, written by hand (Regneri, Koller, and Pinkal 2010; Regneri et al. 2011; Modi and Titov 2013; Frermann, Titov, and Pinkal 2014). We consider the approach of using large text corpora better suited to our goals of learning broad-domain event knowledge, since the events learned are not restricted to those in a hand-constructed corpus. However, some models proposed for ESDs could carry over to our setting. The embeddings of Modi and Titov (2013) in particular are closely related to our best performing system. Their event representation network is similar to the argument composition component of our model, differing in how the representations are trained and treatment of the word representations at the inputs.

McIntyre and Lapata (2009) and McIntyre and Lapata (2010) present, to our knowledge, the only previous work to apply these models to narrative generation. We expect improvements in event prediction accuracy demonstrated here to translate into better results in this downstream task.

Narrative Generation

We adopt the same approach as Chambers and Jurafsky (2008) to extraction of events. They assume that an event is described each time an entity is an argument to a verb. The event is represented by a pair of the verb lemma and the grammatical dependency relation between the verb and the entity (*subj*, *obj* or *iobj*), which we refer to as a *predicate-GR*. An example is shown in figure 1. C&J08 can be applied directly as a component in a narrative generation system (McIntyre and Lapata 2009; 2010). However, the model has some limitations in this context, which we set out to address.

Firstly, in predicting the next event, C&J08 looks only at the co-occurrences of predicate-GRs. Sometimes this contains the most important information about the event – e.g. *(arrest, obj)* – but often the meaning of the event is drastically changed by its arguments – e.g. *(perform play, subj)* vs. *(perform surgery, subj)*. In other cases, almost no information is conveyed by the predicate alone – e.g. *(go on holiday, subj)* vs. *(go to church, subj)*. We address this by incorporating arguments into our representations of events.

Secondly, the model uses only co-occurrence statistics of specific pairs of predicate-GRs: it does not generalize from the observations to assign scores to unobserved pairs. It makes remarkably good judgements about frequent predicates, but is less successful with rare events. For example, a model trained on the Gigaword NYT corpus has little information concerning *underestimate* outside very specific contexts. A model that exploits contextual similarities between, say, *underestimate* and *calculate* may be able to make more informed predictions by making better use of the limited information it has about infrequent predicates. We address this by using continuous vector embeddings of words and events.

One aspect of an event prediction model for narrative generation not considered here is the temporal order of events. For example, we will produce improbable narratives if, presented with *(die, subj)*, our model suggests *(live, subj)* as the next event, simply because the two often co-occur. Building such information into a model is a non-trivial task, since the order of events in a document rarely corresponds to their temporal order. Chambers and Jurafsky first build an unordered model of event associations and then apply a model of partial temporal ordering while building narrative schemas. There is a large body of work on inferring temporal relations (UzZaman et al. 2013), including recent work specifically on ordering of events (Frermann, Titov, and Pinkal 2014; Abend, Cohen, and Steedman 2015). Such models could be used to constrain suggestions made by the models discussed here, an extension we leave to future work.

Evaluating Representations for Prediction

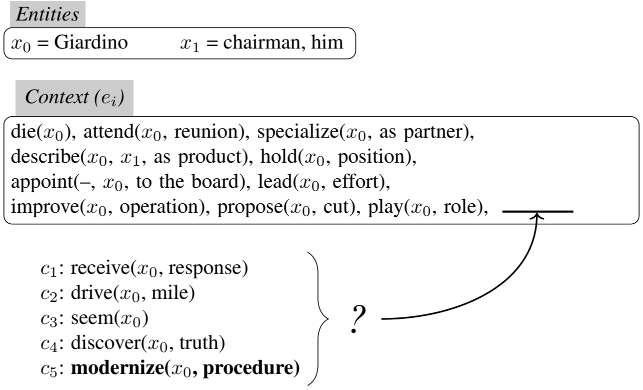
We consider methods for representing information about not just predicate-GRs, but verb arguments in inferring event models. We therefore require an evaluation task capable of comparing models in a context where this richer information is available. The narrative cloze task (Chambers and Jurafsky 2008) evaluates a predictive model by measuring the rank assigned to the observed next event given its context in a chain and a full vocabulary of possible events. When including predicate arguments, the task is problematic: the vocabulary becomes unmanageably large, once, e.g. *put(x, money, in clubhouse)* and *put(x, robber, in jail)* are distinguished, and ranking metrics less meaningful. Recent work has called into question the value of narrative cloze, even for comparing models that represent events only by their predicate-GRs (Rudinger et al. 2015).

We propose a multiple choice version of the narrative cloze task, MCNC, inspired by multiple choice variants of the cloze task for language assessment (Sadeghi 2014). As before, the system is presented with a series of contextual events, $e_0, \dots, e_{n-1}$. It is given five randomly ordered candidates, $c_0, \dots, c_4$, to choose the next event from, one of which is the observed event, the others sampled at random from elsewhere in the corpus. The randomly sampled events have their protagonist replaced by the protagonist of the current chain and any other entities replaced by randomly chosen other entities from the same document as the current chain. An example is shown in figure 2. MCNC allows us to compare models that take account of richer information from the text about both context and candidate events.

An additional advantage of this form of evaluation is that the task can in principle be completed by humans. A human study would be a valuable comparison to the models' results. The test data is imperfect, due to noise in the automatic extraction process and the random sampling of confounders. Our informal initial human studies suggest these are indeed problems, but not so common as to invalidate conclusions drawn here. We have not yet carried out large-scale human annotation, but plan to do so in the future.

Dataset

Following Chambers and Jurafsky (2008; 2009), we extract events from the NYT portion of the Gigaword corpus (Graff et al. 2003). The event extraction pipeline follows an almost identical procedure to Chambers and Jurafsky (2009), using the C&C tools (Curran, Clark, and Bos 2007) for PoS tagging and dependency parsing and OpenNLP[1] for phrase-structure parsing and coreference resolution. In addition to the events inferred from verbs, we also extract predicative adjectives where an entity is an argument to the verb *be* or *become* – e.g. *Giardino was upset* ⇒ *be(x_0, upset)* – with the intention of approximating narrative state information, just as verbs approximate events. Such information is a potentially helpful addition to a model's output for downstream narrative generation. As in C&J08, other occurrences of *be* are dropped.



Entities
x_0 = Giardino
x_1 = chairman, him
Context (e_i)
die(x_0), attend(x_0, reunion), specialize(x_0, as partner),
describe(x_0, x_1, as product), hold(x_0, position),
appoint(–, x_0, to the board), lead(x_0, effort),
improve(x_0, operation), propose(x_0, cut), play(x_0, role),
c_1: receive(x_0, response)
c_2: drive(x_0, mile)
c_3: seem(x_0)
c_4: discover(x_0, truth)
c_5: **modernize(x_0, procedure)**
?

Figure 2: Multiple choice narrative cloze prediction task. The observed event is marked in bold.

As well as the lemmatized verb and its dependency relation to the chain entity, the dataset also includes any subject, object or indirect object arguments to the verb identified by the parser. This may be another coreference entity (as for *describe* in figure 2), or the text of a noun phrase (though our models use just the headword of the noun phrase).

We randomly select 10% of the documents in the corpus to use as a test set and 10% to use as a development set, the latter being used to compare architectures and optimize hyperparameters prior to evaluation. For repeatability, we make the lists of documents selected for each set available online[2].

Using the full set of extracted chains for evaluation suffers from an over-emphasis on frequent predicate types: e.g., the predicate-GR *(have, subj)* accounts for 3.5% of the training set. To mitigate this, we build a *stopevent* list (analogous to *stopwords* – frequent words that carry little meaning), consisting of the 10 most frequent predicate-GRs in the training set and filter events using these out of the chains used for evaluation. The full training set consists of 830,643 documents, with 11,538,312 event chains.

Models

Chambers & Jurafsky 08

We take C&J08 as our starting point. Each event e is represented by its predicate $p(e)$ and dependency relation to the

[1] https://opennlp.apache.org/

[2] http://mark.granroth-wilding.co.uk/\\papers/what_happens_next/

Predicate events: service(x_0, machine), report(x_0), put(x_0, money, in clubhouse), lie+down(x_0), . . .

a. `word2vec` 'sentence': *service:subj report:subj put:subj lie+down:subj*

b. `word2vec` 'sentence' with arguments: *service:subj arg:guards arg:machine report:subj arg:guards put:subj arg:guards arg:money arg:clubhouse lie+down:subj arg:guards*

Figure 3: Example chain from figure 1, in the form in which it is presented to `word2vec`. In **a**, we learn embeddings for predicate-GRs only, in terms of surrounding predicate-GRs. In **b**, we also learn embeddings for argument headwords.

chain's entity $d(e)$ (together a predicate-GR, see figure 1). As a shorthand, we use $pg(e) = (p(e), d(e))$.

For any given pair of event types represented in this way, their PPMI is used as a relatedness score, computed from their co-occurrence in the same chains in the training set. Given the event context in MCNC, we compute a score $s(c)$ for a candidate next event c by summing its PPMI with each of the n context events.

$$s(c) = \sum_{j=0}^{n-1} ppmi(pg(c),\ pg(e_j)) \qquad (1)$$

Bigram

Jans et al. (2012) find that a bigram conditional probability model performs better on the narrative cloze ranking task than C&J08. We evaluate a simple conditional model, BIGRAM, that uses maximum likelihood estimates of the probability of each predicate-GR, conditioned on individual previous events. We apply Laplace smoothing to the estimates and backoff to unigram probabilities for unobserved contexts. The score assigned to a candidate is its average probability given each of the context events.

$$s(c) = \frac{1}{n} \sum_{j=0}^{n-1} P(pg(c) \mid pg(e_j)) \qquad (2)$$

Distributional Vector Model

A shortcoming of C&J08 is that it only assigns a score to pairs of events seen together in the training corpus. Models that represent events in a continuous vector space may give more reliable judgements of how likely two infrequent event types are to co-occur. We use *latent semantic indexing* (LSI, Deerwester et al. 1990) to represent events in terms of the contexts in which they have been seen, as a baseline for other vector-space models.

A matrix is built of co-occurrence counts of predicate-GRs. Each row represents a predicate-GR and each cell the number of times it appears in the same chain as another particular predicate-GR. Singular value decomposition (SVD) on the matrix produces a lower-dimensional, dense representation of each predicate-GR, S. We reduce the dimensionality to 200.

To get a score for each candidate next event, we compute a representation of the context as the sum of the vectors for each predicate-GR. We score each candidate by the cosine similarity of its predicate-GR's vector to the context vector[3]. We refer to this model as DIST-VECS.

$$s(c) = \mathit{cosine}(\sum_{j=0}^{n-1} S_{pg(e_j)}, S_{pg(c)}) \qquad (3)$$

We also experimented with a variety of other methods of deriving vectors for events, among them LDA and an LDA-inspired generative model of predicates and arguments closely related to that of Chambers (2013). Since none of them significantly outperformed DIST-VECS on the development set, we do not discuss them further here.

Word2vec Word Representations

Mikolov et al. (2013a) introduce a method for efficiently learning embeddings (dense vector representations) of words from large text corpora that have proved to be effective at capturing a variety of relations between words and useful for a range of tasks (Mikolov et al. 2013b). They are learned by training a *skip-gram* model, in which surrounding words are predicted based on vector similarity to the current word. The authors' implementation is available in the form of the `word2vec` tool.

Embeddings of verbs learned by `word2vec` could provide a suitable measure to judge the relatedness of two events. E.g., given a context event *criticize(politician, x_0)* and a candidate *repeal(parliament, x_0)*, a close relation between *criticize* and *repeal* in the word vector space could provide evidence that this is a good candidate. This method has the advantage that vectors can be learned for a huge vocabulary by running `word2vec` over a large corpus. It can score almost any pair of events encountered at test time, regardless of whether they appeared together, or even at all, in the event chain training set. It also provides a trivial way to compose predicates with their arguments, since their vector representations live in the same, generic word-meaning space. Mikolov et al. (2013a) found that the vectors compose well for similarity-based tasks under vector addition.

We try a verb-only model and a verb-argument model. In the verb-only model, MIKOLOV-VERB, we represent an event using the vector for its verb, summing context events as before and computing cosine similarity. We use the 300-dimensional vectors W trained by Mikolov et al. (2013b) on the Google News corpus and made available by the authors[4].

$$s(c) = \mathit{cosine}(\sum_{j=0}^{n-1} W_{p(e_j)}, W_{p(c)}) \qquad (4)$$

[3] Levy, Goldberg, and Ramat-Gan (2014) observe that this is equivalent to maximizing the sum of the similarity to each context event.

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

MIKOLOV-VERB+ARG represents each event as the sum of the vectors from W for its verb and each of its arguments.

$$s(c) = \mathit{cosine}(\sum_{j=0}^{n-1} W_{p(e_j)} + W_{a0(e_j)} + W_{a1(e_j)} + W_{a2(e_j)}, \\ W_{p(c)} + W_{a0(c)} + W_{a1(c)} + W_{a2(c)}) \quad (5)$$

Word2vec Event Representations

Another way to use `word2vec` to derive representations of events is to learn embeddings from the event chains. Like the distributional vector model above, the vectors should have the property that events that occur in similar chain contexts are close together.

Vectors are learned by treating each event's predicate-GR as a word and each training chain as a sentence presented to `word2vec`[5] (see figure 3a). We train a skipgram model with hierarchical sampling, using a window size of 5 and vector size of 300. We call this WORD2VEC-PRED.

As a first step to including argument words in the vector representations, WORD2VEC-PRED+ARG, we learn a representation of predicates and arguments together by simply placing all of the words in the 'sentence'. Thus, each predicate and argument functions both as context for surrounding predicates and arguments and as a target word itself (see figure 3b). We increase the window size to 15 to ensure that as many surrounding events are included in the context as before. The representation of an event is now the sum of the vectors for the predicate-GR and each of the arguments.

$$s(c) = \mathit{cosine}(\sum_{j=0}^{n-1} W_{p(e_j):d(e_j)} + W_{\text{arg}:a0(e_j)} + \\ W_{\text{arg}:a1(e_j)} + W_{\text{arg}:a2(e_j)}, \\ W_{p(c):d(c)} + W_{\text{arg}:a0(c)} + W_{\text{arg}:a1(c)} + W_{\text{arg}:a2(c)}) \quad (6)$$

Neural Compositional Representations

We train a neural network, EVENT-COMP, to learn a non-linear composition of predicates and arguments into an event representation, shown in figure 4. As with WORD2VEC-PRED+ARG, it has a large vocabulary of vectors corresponding to predicate-GR and argument words. The vectors corresponding to an event's predicate and each of its argument positions are concatenated to form the first layer representing the event. Zero vectors are used for empty arguments and unseen words. A series of layers (the *argument composition*), each with a tanh activation function, produces a lower-dimensional representation of the event. For a pair of events, a further series of layers (the *event composition*), again with tanh activation functions and a sigmoid activation function on the final layer, produces a single output value, the *coherence score* (*coh*), representing how confident the model is that the two events are from the same chain.

The word vectors are initialized using the vectors learned by WORD2VEC-PRED+ARG. The argument composition layers are initialized by training a stack of denoising autoencoders, so that the initial representation of an event is an efficient compression of the `word2vec`-trained representations of its predicate and arguments. The event composition layers are randomly initialized[6]. Some training iterations are performed updating only the weights of the event composition. Further iterations then update all the weights, including the input word vectors[7]. Although word vectors and argument composition function weights are updated at this stage for both events in a pair, the parameters are tied, so that only a single event representation network is learned, used for events in both positions.

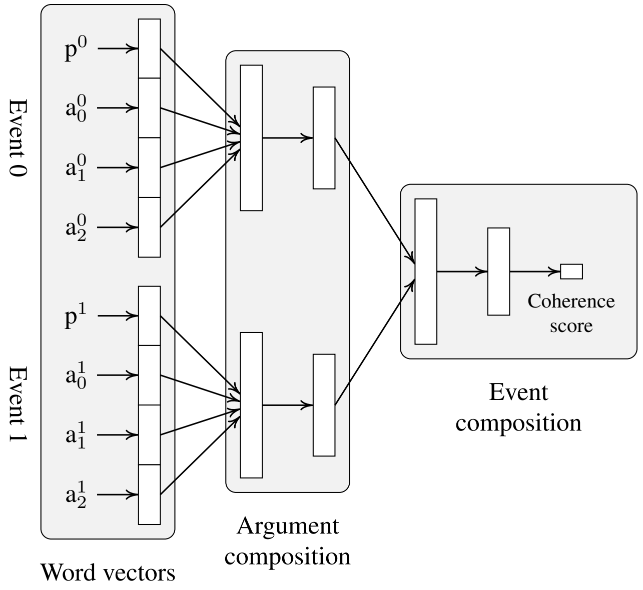


Figure 4: Neural event composition model, which composes each event's predicates and arguments, then scores how strongly they are expected to appear in the same chain.

Both stages minimize the objective function:

$$\frac{1}{m}\sum_{i=1}^{m} -\mathit{log}(p_i \times \mathit{coh}(e0_i, e1_i) + \\ (1 - p_i) \times (1 - \mathit{coh}(e0_i, e1_i)) + \lambda L(\theta) \quad (7)$$

where $p_i = 1$ for positive examples and 0 for negative, $\mathit{coh}(e0_i, e1_i)$ is the score output by the network for the ith training pair and $L(\theta)$ is an l2 regularization term on all weights.

Training the event composition function requires positive and negative samples of pairs of events. For every event $e0_i$ in a chain containing more than one event, another event in the chain is chosen at random to produce a positive sample $(e0_i, e1_i)$. An event is also chosen at random from outside

[5] We use the Gensim implementation: http://radimrehurek.com/gensim/

[6] Autoencoder pretraining of these layers did not improve the results.

[7] We find updating the word vectors at this stage to be beneficial (c.f. Modi and Titov 2013). Unlike Modi and Titov, we learn distinct representations for predicates and each argument slot.

System	Accuracy (%)
Chance baseline	20.00
C&J08	30.52
Bigram	29.67
Dist-vecs	27.94
Mikolov-verb	24.57
Mikolov-verb+arg	28.97
Word2vec-pred	40.17
Word2vec-pred+arg	42.23
Event-comp	**49.57**

Table 1: Model accuracy on the MCNC task.

the current chain to serve as $e1_i$ for a negative sample, which it is assumed should receive a low score.

The input vector for each word is 300-dimensional. We use two hidden layers in the argument composition, with sizes 600 and 300, and two in the event composition, with sizes 400 and 200. Autoencoders were all trained with 30% dropout corruption for 2 iterations over the full training set, with a learning rate of 0.1 and $\lambda = 0.001$. Both subsequent training stages used a learning rate of 0.01 and $\lambda = 0.01$[8]. The first (event composition only) was run for 3 iterations, the second (full network) for 8. All stages of training used SGD with 1,000-sized minibatches.

At test time, the model is used to score candidate next events by averaging pairwise scores with the context events, just as in C&J08:

$$s(c) = \frac{1}{n}\sum_{j=0}^{n-1} coh(c,\ e_j) \tag{8}$$

Results

The test set prediction accuracy of each of the models is shown in table 1. C&J08 performs comfortably above the chance baseline of 20%. Bigram achieves the same level of accuracy as C&J08: the difference is not significant[9]. Dist-vecs, our baseline vector similarity method, does not quite match the performance of C&J08.

Using embeddings for just the verb of each event, Mikolov-verb, performs worse than C&J08, but substantially above chance. Including the embeddings of argument headwords improves the result, almost matching C&J08. Note that neither of these models is trained on event chain data.

The best results of the predicate-GR-only models were produced by using word2vec to learn vector representations of predicate-GRs from event chains (Word2vec-pred). A further gain is produced by also learning a representation of argument headwords, Word2vec-pred+arg, a considerable improvement over C&J08.

Event-comp, which learns a complex composition of arguments and predicates, achieves a further substantial improvement, giving the best result by far of all the models. We attribute this to a combination of the non-linear argument composition function and the learned non-linear combination of the vectors' dimensions to score event pairs.

Implementations of all the models and the evaluation, as well as the evaluation dataset split, are available at http://mark.granroth-wilding.co.uk/\\papers/what_happens_next/.

Conclusion

We consider the problem of automatic acquisition of event knowledge from text as introduced by Chambers and Jurafsky (2008), but with a focus on next event prediction. We introduce a new development of the narrative cloze task (Chambers and Jurafsky 2008): *multiple choice narrative cloze*. We use this task to compare several approaches to event prediction that involve deriving a vector representation of events. We find that the word2vec learning technique of Mikolov et al. (2013a) can be used to induce event representations that, like C&J08, use only the predicate-GRs and achieve considerably better predication accuracy. A simple technique for using argument words to influence the vector representation of an event yields further improvement.

We then use the representations of predicates and their arguments learned in this way as initialization to a neural network model that predicts how likely two events are to appear in the same chain by performing a non-linear composition of their predicates and arguments. At training time, it simultaneously adjusts the word embeddings and learns the predicate-argument composition function and the chain coherence function. This model achieves impressive MCNC prediction accuracy. One possible reason for its success is its ability to capture non-linear interactions between predicates and arguments – e.g. allowing *play golf* and *play dead* to lie in different regions of the vector space. Examination of the best model's performance on infrequent events shows that it is better able than C&J08 to generalize what it has learned about co-occurrence of frequent events to rarer items.

In future work, we plan to experiment with methods to combine whole event chains into a single vector, so the model is not limited to learning associations between pairs of events. We also hope to explore ways in which the model can be used in practice by a narrative generation system for suggesting probable events in a given context.

Acknowledgments

This research was funded by European Commission FP7 framework, through the project WHIM 611560.

References

Abend, O.; Cohen, S. B.; and Steedman, M. 2015. In *Proceedings of NAACL*, 1161–1171. Association for Computational Linguistics.

Balasubramanian, N.; Soderland, S.; Mausam; and Etzioni, O. 2013. Generating coherent event schemas at scale. In

[8] Learning rates, regularization coefficients and network architectures were tuned on the development set.

[9] Significance under Pearson's χ^2 test, $p < 0.05$. All other differences are significant, except Mikolov-verb+arg/Dist-vecs and Mikolov-verb+arg/Bigram.

EMNLP, 1721–1731. Association for Computational Linguistics.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL*, 789–797. Association for Computational Linguistics.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 602–610. Association for Computational Linguistics.

Chambers, N. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of EMNLP*, 1797–1807. Association for Computational Linguistics.

Cheung, J. C. K.; Poon, H.; and Vanderwende, L. 2013. Probabilistic frame induction. *CoRR* abs/1302.4813.

Curran, J.; Clark, S.; and Bos, J. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of ACL, Companion Volume Proceedings of the Demo and Poster Sessions*, 33–36. Association for Computational Linguistics.

Deerwester, S. C.; Dumais, S. T.; Landauer, T. K.; Furnas, G. W.; and Harshman, R. A. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6):391–407.

Frermann, L.; Titov, I.; and Pinkal, M. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. In *Proceedings of EACL*, 49–57. Association for Computational Linguistics.

Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2005. Story plot generation based on CBR. *Knowledge-Based Systems* 18(4):235–242.

Graff, D.; Kong, J.; Chen, K.; and Maeda, K. 2003. English Gigaword, LDC2003T05. *Linguistic Data Consortium, Philadelphia.*

Jans, B.; Bethard, S.; Vulić, I.; and Moens, M. F. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of EACL*, 336–344. Association for Computational Linguistics.

Levy, O.; Goldberg, Y.; and Ramat-Gan, I. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*, 171. Association for Computational Linguistics.

McIntyre, N., and Lapata, M. 2009. Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 217–225. Singapore: Association for Computational Linguistics.

McIntyre, N., and Lapata, M. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of ACL*, 1562–1572. Association for Computational Linguistics.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119. The MIT Press.

Modi, A., and Titov, I. 2013. Learning semantic script knowledge with event embeddings. *CoRR* abs/1312.5198.

Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Pérez y Pérez, R., and Sharples, M. 2004. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-based systems* 17(1):15–29.

Pichotta, K., and Mooney, R. J. 2014. Statistical script learning with multi-argument events. In *Proceedings of EACL*, 220–229. Association for Computational Linguistics.

Regneri, M.; Koller, A.; Ruppenhofer, J.; and Pinkal, M. 2011. Learning script participants from unlabeled data. In *Proceedings of RANLP*, 463–470. Association for Computational Linguistics.

Regneri, M.; Koller, A.; and Pinkal, M. 2010. Learning script knowledge with web experiments. In *Proceedings of ACL 2010*, 979–988. Association for Computational Linguistics.

Rudinger, R.; Rastogi, P.; Ferraro, F.; and Durme, B. V. 2015. Script induction as language modeling. In *Proceedings of EMNLP*, 1681–1686. Association for Computational Linguistics.

Sadeghi, K. 2014. Phrase cloze: A better measure of reading? *The Reading Matrix* 14(1).

Schank, R. C., and Abelson, R. P. 1977. *Scripts, plans, goals and understanding*. Lawrence Erlbaum.

Turner, S. R. 1994. *The creative process: A computer model of storytelling and creativity*. Psychology Press.

UzZaman, N.; Llorens, H.; Derczynski, L.; Allen, J.; Verhagen, M.; and Pustejovsky, J. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 1–9. Association for Computational Linguistics.