Dynamic Entity Representation with Max-pooling Improves Machine Reading

Sosuke Kobayashi and **Ran Tian** and **Naoaki Okazaki** and **Kentaro Inui**
Tohoku University, Japan
{sosuke.k, tianran, okazaki, inui}@ecei.tohoku.ac.jp

Abstract

We propose a novel neural network model for machine reading, *DER Network*, which explicitly implements a reader building dynamic meaning representations for entities by gathering and accumulating information around the entities as it reads a document. Evaluated on a recent large scale dataset (Hermann et al., 2015), our model exhibits better results than previous research, and we find that max-pooling is suited for modeling the accumulation of information on entities. Further analysis suggests that our model can put together multiple pieces of information encoded in different sentences to answer complicated questions. Our code for the model is available at https://github.com/soskek/der-network

1 Introduction

Machine reading systems (Poon et al., 2010; Richardson et al., 2013) can be tested on their ability to answer queries about contents of documents that they read, thus a central problem is how the information of documents should be organized in the system and retrieved by the queries. Recently, large scale datasets of document-query-answer triples have been constructed from online newspaper articles and their summaries (Hermann et al., 2015), by replacing named entities in the summaries with placeholders to form Cloze (Taylor, 1953) style questions (Figure 1). These datasets have enabled training and testing of complicated neural network models of hypothesized machine readers (Hermann et al., 2015; Hill et al., 2015).

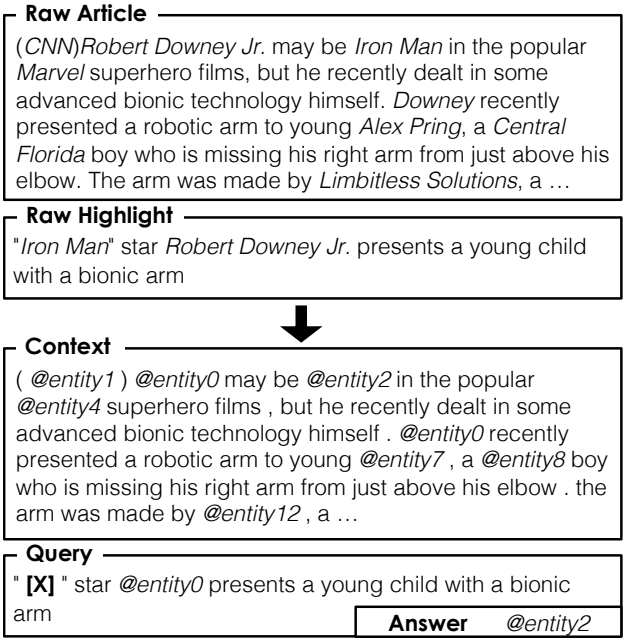

Figure 1: A document-query-answer triple constructed from a news article and its bullet point summary. An entity in the summary (*Robert Downey Jr.*) is replaced by the placeholder **[X]** to form a query. All entities are anonymized to exclude world knowledge and focus on reading comprehension.

In this paper, we hypothesize that a reader without world knowledge can only understand a named entity by dynamically constructing its meaning from the contexts. For example, in Figure 1, a reader reading the sentence "*Robert Downey Jr. may be Iron Man . . .* " can only understand "*Robert Downey Jr.*" as something that "*may be Iron Man*" at this stage, given that it does not know Robert Downey Jr. *a priori*. Information about this entity can only

Proceedings of NAACL-HLT 2016, pages 850–855,
San Diego, California, June 12-17, 2016.

be accumulated by its subsequent occurrence, such as "*Downey recently presented a robotic arm ...*". Thus, named entities basically serve as anchors to link multiple pieces of information encoded in different sentences. This insight has been reflected by the anonymization process in construction of the dataset, in which coreferent entities (e.g. "*Robert Downey Jr.*" and "*Downey*") are replaced by randomly permuted abstract entity markers (e.g. "*@entity0*"), in order to prevent additional world knowledge from being attached to the surface form of the entities (Hermann et al., 2015). We, however, take it as a strong motivation to implement a reader that dynamically builds meaning representations for each entity, by gathering and accumulating information on that entity as it reads a document (Section 2).

Evaluation of our model, *DER Network*, exhibits better results than previous research (Section 3). In particular, we find that max-pooling of entity representations, which is intended to model the accumulation of information on entities, can drastically improve performance. Further analysis suggests that max-pooling can help our model draw multiple pieces of information from different sentences.

2 Model

Following Hermann et al. (2015), our model estimates the conditional probability $p(e|D,q)$, where q is a query and D is a document. A candidate answer for the query is denoted by e, which in this paper is any named entity. Our model can be factorized as:

$$p(e|D,q) \propto \exp(\boldsymbol{v}(e;D,q)^{\mathrm{T}}\boldsymbol{u}(q)) \qquad (1)$$

in which $\boldsymbol{u}(q)$ is the learned meaning for the query and $\boldsymbol{v}(e;D,q)$ the dynamically constructed meaning for an entity, depending on the document D and the query q. We note that (1) is in contrast to the factorization used by Hermann et al. (2015):

$$p(a|D,q) \propto \exp(\boldsymbol{v}(a)^{\mathrm{T}}\boldsymbol{u}(D,q)) \qquad (2)$$

in which a vector $\boldsymbol{u}(D,q)$ is learned to represent the status of a reader after reading a document and a query, and this vector is used to retrieve an answer by coupling with the answer vector $\boldsymbol{v}(a)$.[1]

Factorization (2) relies on the hypothesis that there exists a fixed vector for each candidate answer representing its meaning. However, as we argued in Section 1, an entity surface does not possess meaning; rather, it serves as an anchor to link pieces of information about it. Therefore, we hypothesize that the meaning representation $\boldsymbol{v}(e;D,q)$ of an entity e should be dynamically constructed from its surrounding contexts, and the meanings are "accumulated" through the reader reading the document D. We explain the construction of $\boldsymbol{v}(e;D,q)$ in Section 2.1, and propose a max-pooling process for modeling information accumulation in Section 2.2.

2.1 Dynamic Entity Representation

For any entity e, we take its context c as any sentence that includes a token of e. Then, we use bidirectional single-layer LSTMs (Hochreiter and Schmidhuber, 1997; Graves et al., 2005) to encode c into vectors. LSTM is a neural cell that outputs a vector $\boldsymbol{h}_{c,t}$ for each token t in the sentence c; taking the word vector $\boldsymbol{x}_{c,t}$ of the token as input, each $\boldsymbol{h}_{c,t}$ is calculated recurrently from its precedent vector $\boldsymbol{h}_{c,t-1}$ or $\boldsymbol{h}_{c,t+1}$, depending on the direction of the encoding. Formally, we write forward and backward LSTMs as:

$$\vec{\boldsymbol{h}}_{c,t} = \overrightarrow{LSTM}(\boldsymbol{x}_{c,t}, \vec{\boldsymbol{h}}_{c,t-1}) \quad \text{(forward)} \qquad (3)$$
$$\overleftarrow{\boldsymbol{h}}_{c,t} = \overleftarrow{LSTM}(\boldsymbol{x}_{c,t}, \overleftarrow{\boldsymbol{h}}_{c,t+1}) \quad \text{(backward)} \qquad (4)$$

Then, denoting the length of the sentence c as T and the index of the entity e token as τ, we define the dynamic entity representation $\boldsymbol{d}_{e,c}$ as the concatenation of the vectors $[\vec{\boldsymbol{h}}_{c,T}, \overleftarrow{\boldsymbol{h}}_{c,1}, \vec{\boldsymbol{h}}_{c,\tau}, \overleftarrow{\boldsymbol{h}}_{c,\tau}]$ encoded by a feed-forward layer (Figure 2):

$$\boldsymbol{d}_{e,c} = \tanh(W_{hd}[\vec{\boldsymbol{h}}_{c,T}, \overleftarrow{\boldsymbol{h}}_{c,1}, \vec{\boldsymbol{h}}_{c,\tau}, \overleftarrow{\boldsymbol{h}}_{c,\tau}] + \boldsymbol{b}_d)$$

in which W_{hd} and $\boldsymbol{b}_d$ respectively stand for the learned weight matrix and bias vector of that feed-forward layer. Index hd denotes that W_{hd} is a matrix mapping $\boldsymbol{h}$-vectors to $\boldsymbol{d}$-vectors. Index d shows that $\boldsymbol{b}_d$ has the same dimension as $\boldsymbol{d}$-vectors. We use this convention throughout this paper.

Having $\boldsymbol{d}_{e,c}$ as the dynamic representation of an entity e occurring in context c, we define vector

[1] Hermann et al. (2015) models $p(a|D,q)$ for every word token a in a document. While the approach could be more general because it has the potential to answer other types of questions given appropriate training data, our approach is arguably suitable for the specific task and natural for testing our hypothesis.

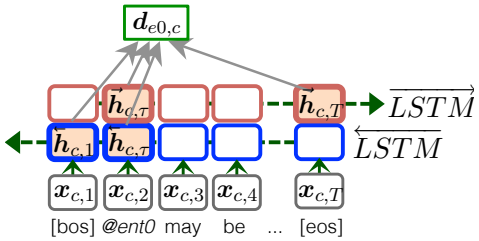

Figure 2: Dynamic entity representation $\boldsymbol{d}_{e,c}$ encodes LSTM outputs, modeling surrounding context.

Figure 3: Max-pooling takes the max value of each dimension of dynamic entity representations, modeling accumulation of context information. It is then fed to $\boldsymbol{x}_{c,\tau}$ as input to LSTMs.

$\boldsymbol{v}(e; D, q)$ for each entity as a weighted sum [2]:

$$\boldsymbol{v}(e; D, q) = W_{dv}\big[\sum_{c \in D} s_{e,c}(q)\boldsymbol{d}_{e,c}\big] + \boldsymbol{b}_v \quad (5)$$

in which $s_{e,c}(q)$ is calculated by the attention mechanism (Bahdanau et al., 2015), modeling the degree to which our reader should attend to a particular occurrence of an entity, given the query q. More precisely, $s_{e,c}(q)$ is defined as the following:

$$s_{e,c}(q) = \frac{\exp(s'_{e,c}(q))}{\sum_{c'} \exp(s'_{e,c'}(q))} \quad (6)$$

$$s'_{e,c'}(q) = \boldsymbol{m}^{\mathrm{T}} \tanh(W_{dm}\boldsymbol{d}_{e,c'} + \boldsymbol{q}) + b_s \quad (7)$$

where $s_{e,c}(q)$ is calculated by taking the softmax of $s'_{e,c'}(q)$, which is calculated from the dynamic entity representation $\boldsymbol{d}_{e,c'}$ and the query vector $\boldsymbol{q}$. The vector $\boldsymbol{m}$, matrix W_{dm}, and the bias b_s in (7) are learned parameters in the attention mechanism. Vector $\boldsymbol{m}$ is used here to map a vector value to a scalar.

The query vector[3] $\boldsymbol{u}(q)$ is constructed similarly as dynamic entity representations, using bidirectional LSTMs[4] to encode the query and then encoding the output vectors. More precisely, if we denote the length of the query as T and the index of the placeholder as τ, the query vector is calculated as:

$$\boldsymbol{u}(q) = W_{hq}[\vec{\boldsymbol{h}}_{q,T}, \overleftarrow{\boldsymbol{h}}_{q,1}, \vec{\boldsymbol{h}}_{q,\tau}, \overleftarrow{\boldsymbol{h}}_{q,\tau}] + \boldsymbol{b}_q \quad (8)$$

Then, $\boldsymbol{v}(e; D, q)$ and $\boldsymbol{u}(q)$ are used in (1) to calculate probability $p(e|D, q)$.

[2] Following a heuristic used in Hill et al. (2015), we add a secondary bias $\boldsymbol{b}'_v$ to $\boldsymbol{v}(e; D, q)$ if the entity e already appears in the query q.

[3] $\boldsymbol{u}(q)$ and another query vector $\boldsymbol{q}$, are calculated respectively, in the same way (8) with unshared model parameters, while sharing the parameters is also promising.

[4] The parameters of the bi-LSTM for queries are not shared with the ones for entity contexts.

2.2 Max-pooling

We expect the dynamic entity representation to capture information about an entity mentioned in a sentence. However, as an entity occurs multiple times in a document, information is accumulated as subsequent occurrences of the entity draw information from previous mentions. For example, in Figure 1, the first sentence mentioning "*Robert Downey Jr.*" relates *Downey* to *Iron Man*, whereas a subsequent mention of "*Downey*" also relates him to a robotic arm. Both of the two pieces of information are necessary to answer the query "*Iron Man star [X] presents ... with a bionic arm*". Therefore, the dynamic entity representations as constructed individually from single sentences may not provide enough information for our reader model. We thus propose the use of max-pooling to model information accumulation of dynamic entity representations.

More precisely, for each entity e, max-pooling takes the max value of each dimension of the vectors $\boldsymbol{d}_{e,c'}$ from all preceding contexts c' (Figure 3). Then, in a subsequent sentence c where the entity occurs again at index τ, we use the vector

$$\boldsymbol{x}_{c,\tau} = W_{dx} \underset{c' \prec c}{\text{max-pooling}}(\boldsymbol{d}_{e,c'}) + \boldsymbol{b}_x$$

as input for the LSTMs in (3) and (4) for encoding the context. This vector $\boldsymbol{x}_{c,\tau}$ draws information from preceding contexts, and is regarded as the meaning of the entity e that the reader understands so far, before reading the sentence c. It is used in place of a vector previously randomly initialized as a notion of e, in the construction of the new dynamic entity representation $\boldsymbol{d}_{e,c}$.

3 Evaluation

We use the CNN-QA dataset (Hermann et al., 2015) for evaluating our model's ability to answer questions about named entities. The dataset consists of (D, q, e)-triples, where the document D is taken from online news articles, and the query q is formed by hiding a named entity e in a summarizing bullet point of the document (Figure 1). The training set has 90k articles and 380k queries, and both validation and test sets have 1k articles and 3k queries. An average article has about 25 entities and 700 word tokens. One trains a machine reading system on the data by maximizing likelihood of correct answers. We use `Chainer`[5] (Tokui et al., 2015) to implement our model[6].

Experimental Settings Named entities in CNN-QA are already recognized. For preprocessing, we segment sentences at punctuation marks ".", "!", and "?".[7] We train our model[8] with hyper-parameters lightly tuned on the validation set[9], and we conduct ablation test on several techniques that improve our basic model.

Results As shown in Table 1, Max-pooling described in Section 2.2 drastically improves performance, showing the effect of accumulating information on entities. Another technique, called "Byway", is based on the observation that the attention mechanism (5) must always promote some entity occurrences (since all weights sum to 1), which could be difficult if the entity does not answer the query. To counter this, we make an artificial occurrence for each entity with no contexts, which serves as a byway to attend when no other occurrences can be reasonably related to the query. This simple trick shows clear effects, suggesting that the attention mechanism plays a key role in our model. Combining these two techniques helps more. Further, we note that initializing our model with pre-trained word vectors[10] is helpful, though world knowledge of entities has been prevented by the anonymization process. This suggests that pre-trained word vectors may still bring extra linguistic knowledge encoded in ordinary words. Finally, we note that our model, full *DER Network*, shows the best results compared to several previous reader models (Hermann et al., 2015; Hill et al., 2015), endorsing our approach as promising. The 99% confidence intervals of the results of full DER Network and the one initialized by word2vec on the test set were $[0.700, 0.740]$ and $[0.708, 0.749]$, respectively (measured by bootstrap tests).

Models	Valid	Test
Basic Proposed Model (Basic)	0.614	0.623
Basic + Max-pooling	**0.712**	0.707
Basic + Byway	0.691	0.706
Basic + Byway, Max-pooling (Full)	0.708	**0.720**
Full + w2v-initialization	**0.713**	**0.729**
Deep LSTMs*	0.550	0.570
Attentive Reader*	0.616	0.630
Impatient Reader*	0.618	0.638
Memory Networks**	0.635	0.684
+ Ensemble (11 models)**	0.662	0.694

Table 1: Accuracy on CNN-QA dataset. Results marked by $*$ are cited from Hermann et al. (2015) and $**$ from Hill et al. (2015).

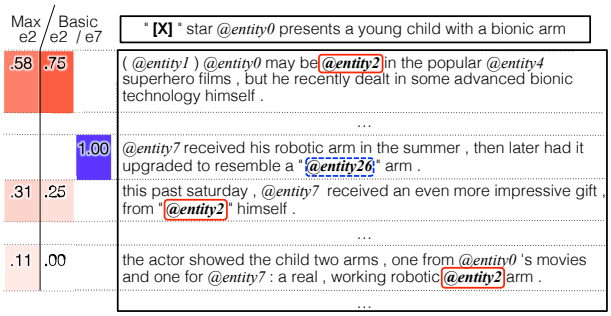

Figure 4: A correct answer found by max-pooling. Attention to each entity occurrence shown on left.

[5] `http://chainer.org/`

[6] The implementation is available at `https://github.com/soskek/der-network`.

[7] Text in CNN-QA are tokenized without any sentence segmentations.

[8] Training process takes roughly a week (3-5 passes of the training data) on a 6-core 2.4GHz Xeon CPU.

[9] Vector dimension: 300, Dropout: 0.3, Batch: 50, Optimization: RMSProp with momentum (Tieleman and Hinton, 2012; Graves, 2013) (momentum: 0.9, decay: 0.95), Learning rate: 1e-4 divided by 2.0 per epoch, Gradient clipping factor: 10. We initialize word vectors by uniform distribution [-0.05, 0.05], and other matrix parameters by Gaussians of mean 0 and variance $2/(\#\text{ rows} + \#\text{ columns})$.

[10] We use GoogleNews vectors from `http://code.google.com/p/word2vec/` (Mikolov et al., 2013).

Analysis In the example shown in Figure 4, our basic model missed by paying little attention to the second and third sentences, probably because it does not mention @*entity0* (*Downey*). In contrast, max-pooling of @*entity2* (*Iron Man*) draws attention to the second and third sentences because *Iron Man* is said related to *Downey* in the first sentence. This helps *Iron Man* surpass @*entity26* (*Transformers*), which is the name of a different movie series in which robots appear but *Downey* doesn't. Quantitatively, in the 479 samples in test set correctly answered by max-pooling but missed by basic model, the average occurrences of answer entities (8.0) is higher than the one (7.2) in the 1782 samples correctly answered by both models. This suggests that max-pooling especially helps samples with more entity mentions.

4 Discussion

It is actually a surprise for us that deep learning models, despite their vast amount of parameters, seem able to learn as intended by the designers. This also indicates a potential that additional linguistic intuitions modeled by deep learning methods can improve performances, as in the other work using max-pooling (LeCun et al., 1998; Socher et al., 2011; Le et al., 2012; Collobert et al., 2011; Kalchbrenner et al., 2014), attention (Bahdanau et al., 2015; Luong et al., 2015; Xu et al., 2015; Rush et al., 2015), etc. In this work, we have focused on modeling a reader that dynamically builds meanings for entities. We believe the methodology can be inspiring to other problems as well.

Acknowledgments

This work was supported by CREST, JST and JSPS KAKENHI Grant Number 15H01702 and 15H05318. We would like to thank members of Preferred Infrastructure, Inc. and Preferred Networks, Inc. for useful discussions. We also thank the anonymous reviewers for comments on earlier version of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, pages 799–804.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1684–1692.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *CoRR*, abs/1511.02301.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665.

Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng. 2012. Building high-level features using large scale unsupervised learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 81–88.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Hoifung Poon, Janara Christensen, Pedro Domingos, Oren Etzioni, Raphael Hoffmann, Chloe Kiddon, Thomas Lin, Xiao Ling, Mausam, Alan Ritter, Stefan Schoenmackers, Stephen Soderland, Dan Weld, Fei Wu, and Congle Zhang. 2010. Machine reading at the university of washington. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 87–95.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*, pages 801–809.

Wilson L. Taylor. 1953. "cloze procedure": a new tool for measuring readability. *Journalism Quarterly*, 30:415–433.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5 - msprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning.*

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The 29th Annual Conference on Neural Information Processing Systems*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2048–2057.