NEAL: A Neurally Enhanced Approach to Linking Citation and Reference

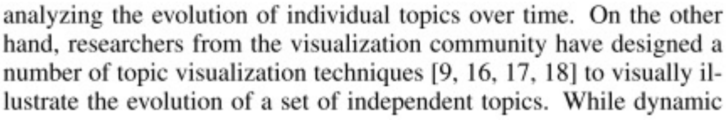
Tadashi Nomoto

[1] National Institute of Japanese Literature
[2] The Graduate University of Advanced Studies (SOKENDAI)
nomoto@acm.org

Abstract. As a way to tackle Task 1A in CL-SciSumm 2016, we introduce a composite model consisting of TFIDF and Neural Network (NN), the latter being a adaptation of the embedding model originally proposed for the Q/A domain [2, 7]. We discuss an experiment using a development data, results thereof, and some remaining issues.

1 Introduction

This paper provides an overview of our efforts to tackle Task 1A at CL-SciSumm 2016, whose stated goal is to locate part of a reference paper (RP) most relevant to a given citation made by a citing paper (CP). To give an idea of what it is about, consider Figure 1.



analyzing the evolution of individual topics over time. On the other hand, researchers from the visualization community have designed a number of topic visualization techniques [9, 16, 17, 18] to visually illustrate the evolution of a set of independent topics. While dynamic

Fig. 1. An example of citation in a scientific publication [3].

In it, you have a sentence that reads:

> On the other hand, researchers from the visualization community have designed to a number of topic visualization techniques [9,16,17,18] ...

Your job is to find passages in the relevant literature (what the authors call 9, 16, 17, and 18), which are most pertinent to the sentence in question. (We denote a passage in referred-to papers linked with a citation by a *citation target* or simply *target*, below and throughout the paper.)

As a way to solve the task, we work with a hybrid of two models: one that is based on TFIDF and another on a single layer Neural Network (NN). Formally, the present approach looks like the following.

$$\sigma(d, r) = \lambda h(d, r) + (1 - \lambda) t(d, r) \tag{1}$$

where h represents a neural network and t a TFIDF based model; d is a citation instance and r a sentence in RP.[3] For a given citation instance d, we rank every sentence r in RP in accordance with σ (while dismissing those with two or less words) and select *two* highest ranked sentences as a target for d. We then remove redundancies in the output with an MMR-like measure: we take a candidate sentence off the output if its similarity with those preceding it exceeds a certain threshold (γ). Thus, the number of the output sentences will be further cut down to one in case they are found to contain redundancies. In the final run, we set γ to 0.24 and λ to 0.1. We call the current setup as 'a neurally enhanced approach to linking citation and reference,' or NEAL for short. Our adding the TFIDF component to NN in σ is meant to compensate for the latter's inability to handle exact word matches effectively due to the low dimensionality of hidden layers into which word features are mapped [9].

One significant consequence of using NN is that it will relieve us from the drudgery of contriving every feature that one needs to train a classifier on: NN learns by itself whatever feature it finds necessary to satisfy an objective function.

In what follows, we discuss the NN portion of σ, which is basically an adaptation of the neural embedding models [2, 1, 7, 8] to the current task. We built the TFIDF part based on statistics collected from the final test data that CL-SciSumm 2016 released.

2 Predicting Similarity with Neural Network

The job of NN is to provides a scoring function h that favors a true target over a false one: that is, to build a function that ensures that $h(d, r^+) > h(d, r^-)$, where r^+ denotes a true target (a sentence humans judged as a target) and r^- a false target (i.e., a sentence not selected as a target). We define h by:

$$h(d, r) = \mathbf{G}(d)^\top \mathbf{F}(r), \tag{2}$$

where $\mathbf{G}(d)$ denotes a vector derived from d and $\mathbf{F}(r)$ a vector from r, through word embedding. In order for d's similarity with its true target (r^+) to be always higher than that with a false target (r^-) [2, 7], we require the following constraint hold for $\mathbf{G}(d)$ and $\mathbf{F}(r)$:

$$\forall_{i,j} \;\; \mathbf{G}(d_i)^\top \mathbf{F}(r_j^+) > 0.1 + \mathbf{G}(d_i)^\top \mathbf{F}(r_j^-),$$ [4]

which is tantamount to:

$$\text{minimize:} [0.1 - \mathbf{G}(d_i)^\top \mathbf{F}(r_j^+) + \mathbf{G}(d_i)^\top \mathbf{F}(r_j^-)]_+,$$

Figure 2 gives a general picture of how we arrive at $\mathbf{G}(d)^\top \mathbf{F}(r)$. We start at the bottom, with inputs that represent d and r. We initially translate every word in a citation instance and target into word indices ranging from 0 to 15,456, which will be assembled into a binary vector, where the presence or absence of word is marked with 1 or 0 at an

[3] t is defined as: $t(d, r) = \dfrac{\mathbf{d} \cdot \mathbf{r}}{\|\mathbf{d}\| \, \|\mathbf{r}\|}$, where $\mathbf{d}$ and $\mathbf{r}$ are a vector of TFIDF weights representing d and r, respectively.

[4] '0.1' represents a margin we have taken from [2].

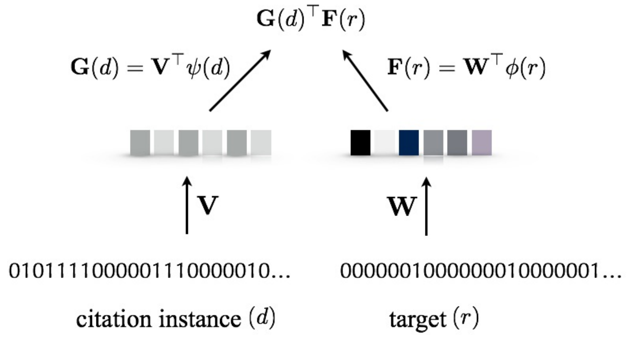

$\mathbf{G}(d)^\top\mathbf{F}(r)$
$\mathbf{G}(d) = \mathbf{V}^\top\psi(d)$
$\mathbf{F}(r) = \mathbf{W}^\top\phi(r)$
V
W
0101111000001110000010...
0000001000000010000001...
citation instance (d)
target (r)

Fig. 2. Predicting Citation/Target Similarity Through Embedding

index assigned to it: thus having a 1 at the i-th unit means that a relevant input sentence contains a word indexed with i. We denote a binary vector for d so derived by $\psi(d)$ and that for r by $\phi(r)$. $[x]_+$ is a positive part of x.

We project $\psi(d)$ and $\phi(r)$ into two hidden layers, $\mathbf{V}$ and $\mathbf{W}$, through word embedding. $\mathbf{V}$ is an $N_v \times K$ matrix ($\in \mathbb{R}^{N_v \times K}$) and $\mathbf{W}$ an $N_e \times K$ matrix ($\in \mathbb{R}^{N_e \times K}$), with N_v, N_e, and K indicating lengths of $\psi(d)$, $\phi(r)$ and a hidden layer, respectively. (In the test run, we set $K = 30$ and $N_v = N_e = 15,457$.) Now we let

$$\mathbf{G}(d) = \mathbf{V}^\top\psi(d),$$

and

$$\mathbf{F}(r) = \mathbf{W}^\top\phi(r).$$

To determine values in $\mathbf{V}$ and $\mathbf{W}$, we launch an iterative training process. Suppose that we have the training data consisting of triples,

$$D = \{(d_i, r_i^+, S_i^-)\}_{i=1\ldots m}$$

with d_i representing a citing instance, r_i^+ a true target and $S_i^- = \{r_{i1}^-, \ldots, r_{in}^-\}$ indicating a set of false targets for d_i. For each $(d_i, r_i^+, S_i^-) \in D$, we do the following.

1. For each $r_i^- \in S_i^-$, perform a stochastic gradient descent (SGD) to minimize:

$$[0.1 - \mathbf{G}(d_i)^\top\mathbf{F}(r_i^+) + \mathbf{G}(d_i)^\top\mathbf{F}(r_i^-)]_+$$

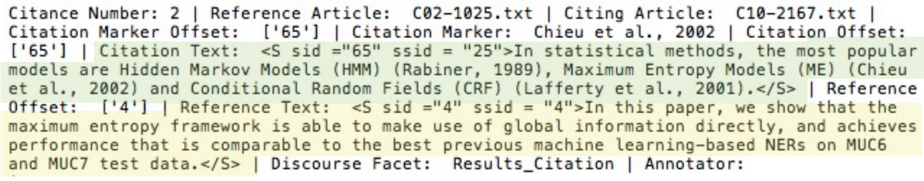
2. Ensure that columns of $\mathbf{W}$ and $\mathbf{V}$ are all normalized.

We developed training data from the 'Development-Set-Apr8' dataset (henceforth, DSA) [5], which produced 4,608 training instances. We trained the model over 10 epochs, meaning that it went through 46,080 training instances. We performed SGD using an optimization algorithm known as AdaGrad-RDA, a regularized version of AdaGrad [4].[5]

[5] [6] developed a variant of LSTM to address an essentially same problem as discussed here, which could serve as a possible replacement of the embedding model the present model employs.

3 Evaluation

Prior to the actual run, we conducted an experiment using DSA to see how well NEAL works. The dataset comes with ten topic clusters, each of which consists of one reference paper and a number of papers that contain citations to that paper. Following a leave-one-out cross validation scheme, we split DSA into two blocks, one containing nine topic (RP) clusters and the other one. We used the former for training NEAL (or h, to be precise), and tested it out on the remaining block. The performance was measured by ROUGE-LCS, which produces the normalized length of a longest, possibly discontiguous, string of words shared by predicted and true targets. Figure 3 illustrates a citation and a corresponding target (made available by CL-SciSumm 2016 as part of gold standard data). The area shaded in green represents a citation in CP and one in yellow a target in RP. A citation and a target can span an arbitrary number of sentences.



```
Citance Number: 2 | Reference Article:  C02-1025.txt | Citing Article:  C10-2167.txt |
Citation Marker Offset:  ['65'] | Citation Marker:  Chieu et al., 2002 | Citation Offset:
['65'] | Citation Text:  <S sid ="65" ssid = "25">In statistical methods, the most popular
models are Hidden Markov Models (HMM) (Rabiner, 1989), Maximum Entropy Models (ME) (Chieu
et al., 2002) and Conditional Random Fields (CRF) (Lafferty et al., 2001).</S> | Reference
Offset:  ['4'] | Reference Text:  <S sid ="4" ssid = "4">In this paper, we show that the
maximum entropy framework is able to make use of global information directly, and achieves
performance that is comparable to the best previous machine learning-based NERs on MUC6
and MUC7 test data.</S> | Discourse Facet:  Results_Citation | Annotator:
```

Fig. 3. Citation and Target

Table 1. Dry-Run Test Set (DSA)

RP	#CP	$\|D\|$	$\|E\|$	$\|T\|$	RP	#CP	$\|D\|$	$\|E\|$	$\|T\|$
C02-1025	18	4,142	8,284	23	N04-1038	20	4,190	8,340	24
C08-1098	22	3,824	7,648	29	P06-2124	12	4,367	8,734	18
C10-1045	13	3,621	7,242	33	W04-0213	13	4,327	8,654	18
D10-1083	11	4,329	8,658	18	W08-2222	9	4,519	9,038	9
E09-2008	10	4,526	9,052	8	W95-0104	25	3,413	6,826	39

Some statistics on DSA are shown in Table 1. RP refers to a reference paper, #CP the count of relevant citing papers, $|D|$ the the number of instances used for training. $|E|$ indicates how many instances are processed over the entire span of epochs, and $|T|$ the number of citation-target pairs we used to test NEAL. Term and document frequencies (to be used for $t(\cdot, \cdot)$ in σ) were collected from DSA. K, N_v and N_e $-$ parameters that define the shape of NN $-$ were set to $30, 15, 457$, and $15, 457$ (we also used the settings for the final run).

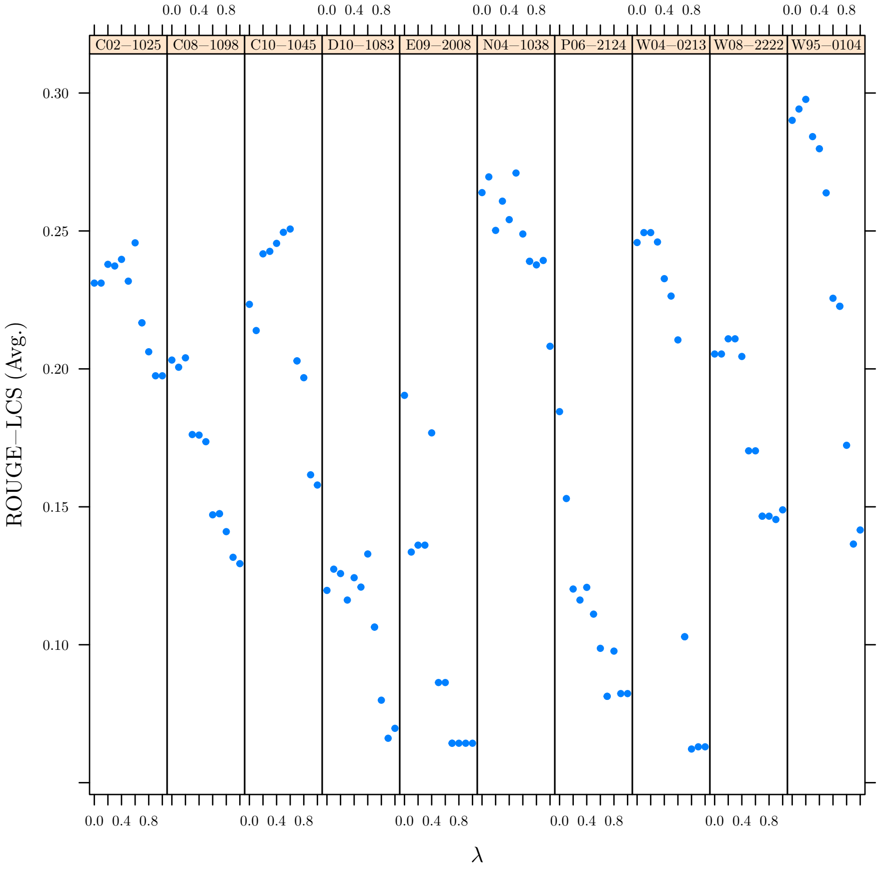


Fig. 4. Plot of Performance vs. λ for DSA. The title of each strip (e.g. C02-1025) represents a designator for a given reference paper, which has 9 to 25 citing papers (Table 1).

The test proceeded as follows. For a given pair c and t of citation and target, we rank each sentence r in RP in accordance to $\sigma(c, r)$ and select top one or two candidates as a possible target (call it g). We then determine ROUGE-LCS for g and its true target t, average scores over a entire set of citation-target pairs that belong to a particular topic cluster. Our computation of ROUGE-LCS, however, did not include tokens with less than 5 characters and those with more than 9 characters, as they were often found to be garbled and unintelligible. We also chose not to use stemming or filter out stop words.

Figure 4 shows by cluster performance of NEAL. The horizontal axis denotes the value of λ and the vertical axis ROUGE-LCS scores. That λ affects the overall performance is clearly seen. Note that NEAL reduces to TFIDF at $\lambda = 0$, and turns into a full-fledged NN at $\lambda = 1$. Thus if NEAL's performance peaks at $\lambda > 0$, it will mean that NN-enabled NEAL performs better than TFIDF, or else is just as good as the latter. We observe in Figure 4 a general tendency for the performance to climb highest somewhere between 0 and 1, suggesting the superiority of NN over TFIDF, although there are notable exceptions at E09-2208 and P06-2124 where the score peaks at $\lambda = 0$.

Table 2. Peak Performance (PP) and λ

RP	PP	λ	RP	PP	λ
C02-1025	0.2457	0.6	N04-1038	0.2710	0.5
C08-1098	0.2040	0.2	P06-2124	0.1845	0.0
C10-1045	0.2507	0.6	W04-0213	0.2494	0.2
D10-1083	0.1329	0.6	W08-2222	0.2109	0.3
E09-2008	0.1904	0.0	W95-0104	0.2977	0.2

Table 2 lists values of λ at which we had peak performance for each of the topic clusters. 8 out of 10 clusters had peak performance at $\lambda > 0$, demonstrating that enabling NN generally leads to a gain in performance.

4 Final Remarks

We have presented what we call a 'neurally enhanced approach to linking citation and reference' or NEAL, describing in some detail what machinery is involved and what we found in an experiment with the development data. The results appear to suggest a moderate impact of the neural network (NN) on the overall performance. But NEAL's performance against TFIDF is far from impressive. We suspect that its somewhat lackluster performance may have been caused by our inability to clearly demarcate true and false targets: there are some words that appear both in true and false targets, which could easily derail the classifier.

Moreover, one could argue that the results of our experiment with DSA substantiated a concern that [9] expressed about NN's handling of word matches: at $\lambda = 1$ when NN was decoupled from TFIDF completely, its performance plummeted to the ground.

As a way out, [9] suggests that we use the following instead of Equation (2).

$$h(d,r) = \mathbf{G}(d)^\top \mathbf{F}'(r,d)$$

$\mathbf{F}'$ is an $\mathbf{F}$ conditioned on d, where you turn off all the words in $\phi(r)$ that are not found in $\psi(d)$. What makes the idea interesting is that it points to a possibility of embedding t into h by slightly modifying the way we build $\phi(r)$ and $\psi(d)$. While it is not clear at the moment how it plays out, we believe that the idea is definitely worth a try, and something we like to explore in the future work.

References

1. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data. N pp. 1–9 (2013)
2. Bordes, A., Weston, J., Usunier, N.: Supervised Embedding Models. In: ECML PKDD 2014 (2014)
3. Cui, W., Liu, S., Tan, L., Shi, C., Song, Y., Gao, Z., Tong, X., Qu, H.: Textflow: Towards better understanding of evolving topics in text. IEEE Transactions on Visualization and Computer Graphics 17(12), 2412–2421 (2011)
4. Duchi, J.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization . Journal of Machine Learning Research 12, 2121–2159 (2011)
5. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., Kan, M.Y.: Overview of the 2nd computational linguistics scientific document summarization shared task (cl-scisumm 2016). In: The Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016). Newark, New Jersey, USA (2016)
6. Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., Ward, R.: Deep Sentence Embedding Using the Long Short-Term Memory Networks. In: Proceedings of the 31st International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. (2015), `http://arxiv.org/abs/1502.06922`
7. Weston, J., Bengio, S., Usunier, N.: Large scale image annotation: Learning to rank with joint word-image embeddings. Machine Learning 81, 21–35 (2010)
8. Weston, J., Bordes, A., Yakhnenko, O., Usunier, N.: Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. Empirical Methods in Natural Language Processing (October), 1366–1371 (2013), `http://aclweb.org/anthology/D/D13/D13-1136.pdf`
9. Weston, J., Chopra, S., Bordes, A.: Memory Networks. International Conference on Learning Representations pp. 1–14 (2015), `http://arxiv.org/abs/1410.3916`