PaperRobot: Incremental Draft Generation of Scientific Ideas

Qingyun Wang[1], **Lifu Huang**[1], **Zhiying Jiang**[1],
Kevin Knight[2], **Heng Ji**[1,3], **Mohit Bansal**[4], **Yi Luan**[5]
[1] Rensselaer Polytechnic Institute [2] DiDi Labs [3] University of Illinois at Urbana-Champaign
[4] University of North Carolina at Chapel Hill [5] University of Washington
kevinknight@didiglobal.com, hengji@illinois.edu

Abstract

We present a *PaperRobot* who performs as an automatic research assistant by (1) conducting deep understanding of a large collection of human-written papers in a target domain and constructing comprehensive background knowledge graphs (KGs); (2) creating new ideas by predicting links from the background KGs, by combining graph attention and contextual text attention; (3) incrementally writing some key elements of a new paper based on memory-attention networks: from the input title along with predicted related entities to generate a paper abstract, from the abstract to generate conclusion and future work, and finally from future work to generate a title for a follow-on paper. Turing Tests, where a biomedical domain expert is asked to compare a system output and a human-authored string, show *PaperRobot* generated abstracts, conclusion and future work sections, and new titles are chosen over human-written ones up to 30%, 24% and 12% of the time, respectively.[1]

1 Introduction

Our ambitious goal is to speed up scientific discovery and production by building a *PaperRobot*, who addresses three main tasks as follows.

Read Existing Papers. Scientists now find it difficult to keep up with the overwhelming amount of papers. For example, in the biomedical domain, on average more than 500K papers are published every year[2], and more than 1.2 million new papers are published in 2016 alone, bringing the total number of papers to over 26 million (Van Noorden, 2014). However, human's reading ability keeps almost the same across years. In 2012, US scientists estimated that they read, on average, only 264 papers per year (1 out of 5000 available papers), which is, statistically, not different from what they reported in an identical survey last conducted in 2005. *PaperRobot* automatically reads existing papers to build background knowledge graphs (KGs), in which nodes are entities/concepts and edges are the relations between these entities (Section 2.2).

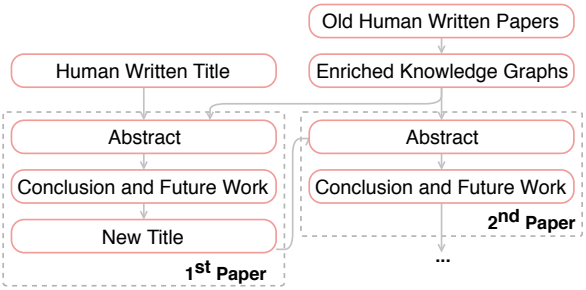

Figure 1: PaperRobot Incremental Writing

Create New Ideas. Scientific discovery can be considered as creating new nodes or links in the knowledge graphs. Creating new nodes usually means discovering new entities (e.g., new proteins) through a series of real laboratory experiments, which is probably too difficult for *PaperRobot*. In contrast, creating new edges is easier to automate using the background knowledge graph as the starting point. Foster et al. (2015) shows that more than 60% of 6.4 million papers in biomedicine and chemistry are about incremental work. This inspires us to automate the incremental creation of new ideas and hypotheses by predicting new links in background KGs. In fact, when there is more data available, we can construct larger and richer background KGs for more reliable link prediction. Recent work (Ji et al., 2015b) successfully mines strong relevance between drugs and diseases from biomedical pa-

[1]The programs, data and resources are publicly available for research purpose at: https://github.com/EagleW/PaperRobot

[2]http://dan.corlan.net/medline-trend/language/absolute.html

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1980–1991
Florence, Italy, July 28 - August 2, 2019.

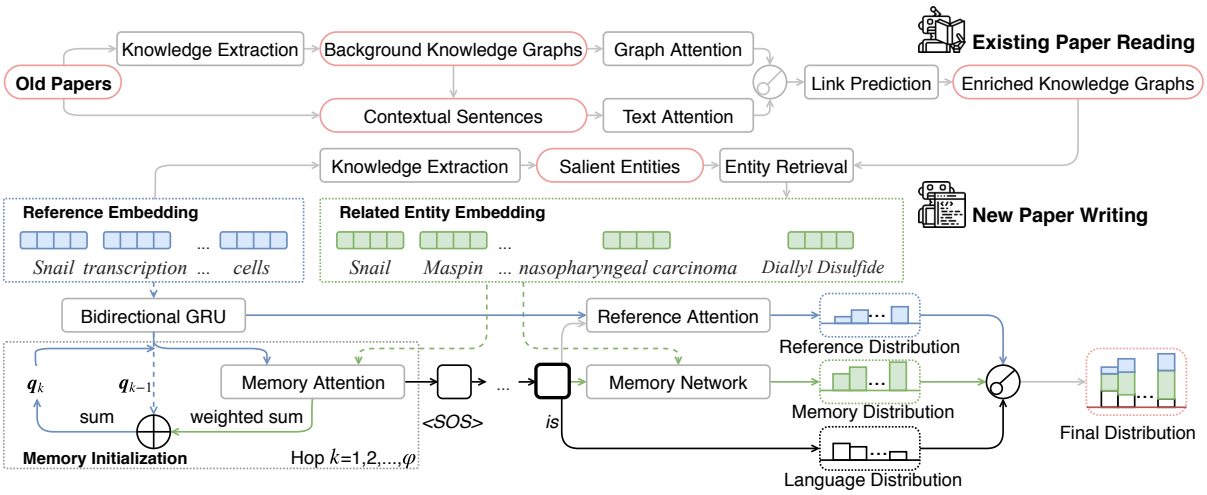

Figure 2: PaperRobot Architecture Overview

pers based on KGs constructed from weighted co-occurrence. We propose a new entity representation that combines KG structure and unstructured contextual text for link prediction (Section 2.3).

Write a New Paper about New Ideas. The final step is to communicate the new ideas to the reader clearly, which is a very difficult thing to do; many scientists are, in fact, bad writers (Pinker, 2014). Using a novel memory-attention network architecture, *PaperRobot* automatically writes a new paper abstract about an input title along with predicted related entities, then further writes conclusion and future work based on the abstract, and finally predicts a new title for a future follow-on paper, as shown in Figure 1 (Section 2.4).

We choose biomedical science as our target domain due to the sheer volume of available papers. Turing tests show that *PaperRobot*-generated output strings are sometimes chosen over human-written ones; and most paper abstracts only require minimal edits from domain experts to become highly informative and coherent.

2 Approach

2.1 Overview

The overall framework of *PaperRobot* is illustrated in Figure 2. A walk-through example produced from this whole process is shown in Table 1. In the following subsections, we will elaborate on the algorithms for each step.

2.2 Background Knowledge Extraction

From a massive collection of existing biomedical papers, we extract entities and their relations to construct background knowledge graphs (KGs). We apply an entity mention extraction and linking system (Wei et al., 2013) to extract mentions of three entity types (**Disease**, **Chemical** and **Gene**) which are the core data categories in the Comparative Toxicogenomics Database (CTD) (Davis et al., 2016), and obtain a Medical Subject Headings (MeSH) Unique ID for each mention. Based on the MeSH Unique IDs, we further link all entities to the CTD and extract 133 subtypes of relations such as **Marker/Mechanism**, **Therapeutic**, and **Increase Expression**. Figure 3 shows an example.

2.3 Link Prediction

After constructing the initial KGs from existing papers, we perform link prediction to enrich them. Both contextual text information and graph structure are important to represent an entity, thus we combine them to generate a rich representation for each entity. Based on the entity representations, we determine whether any two entities are semantically similar, and if so, we propagate the neighbors of one entity to the other. For example, in Figure 3, because *Calcium* and *Zinc* are similar in terms of contextual text information and graph structure, we predict two new neighbors for *Calcium*: *CD14 molecule* and *neuropilin 2* which are neighbors of *Zinc* in the initial KGs.

We formulate the initial KGs as a list of tuples numbered from 0 to κ. Each tuple (e_i^h, r_i, e_i^t) is composed of a head entity e_i^h, a tail entity e_i^t, and their relation r_i. Each entity e_i may be involved in multiple tuples and its one-hop connected neighbors are denoted as $N_{e_i} = [n_{i1}, n_{i2}, ...]$. e_i is

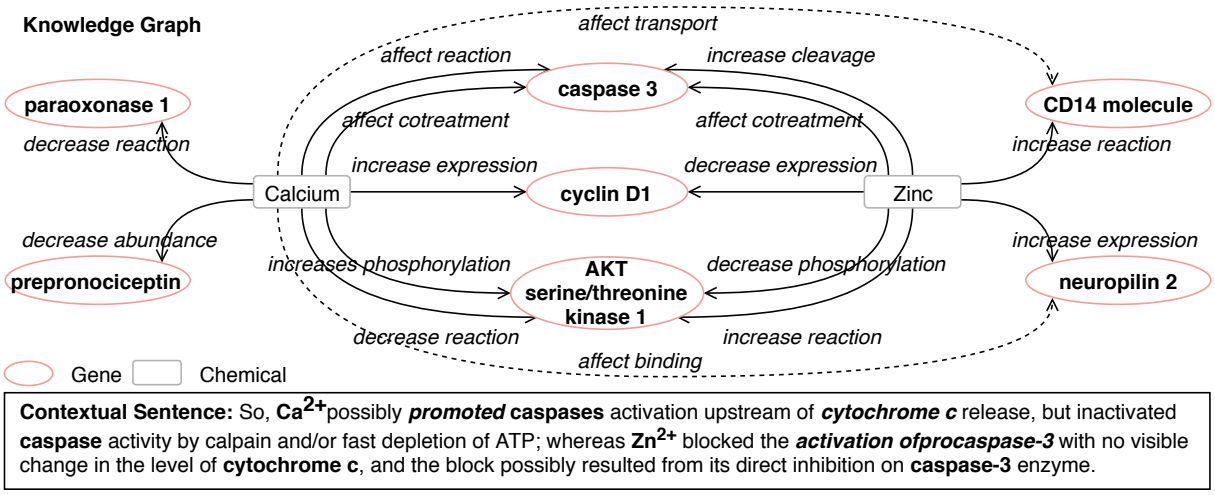

Figure 3: Biomedical Knowledge Extraction and Link Prediction Example (dash lines are predicted links)

also associated with a context description s_i which is randomly selected from the sentences where e_i occurs. We randomly initialize vector representations $\boldsymbol{e_i}$ and $\boldsymbol{r_i}$ for e_i and r_i respectively.

Graph Structure Encoder To capture the importance of each neighbor's feature to e_i, we perform self-attention (Veličković et al., 2018) and compute a weight distribution over N_{e_i}:

$$\boldsymbol{e}_i^{'} = \boldsymbol{W}_e \boldsymbol{e}_i, \quad \boldsymbol{n}_{ij}^{'} = \boldsymbol{W}_e \boldsymbol{n}_{ij}$$
$$c_{ij} = \text{LeakyReLU}(\boldsymbol{W}_f(\boldsymbol{e}_i^{'} \oplus \boldsymbol{n}_{ij}^{'}))$$
$$\boldsymbol{c}_i^{'} = \text{Softmax}(\boldsymbol{c}_i)$$

where $\boldsymbol{W}_e$ is a linear transformation matrix applied to each entity. $\boldsymbol{W}_f$ is the parameter for a single layer feedforward network. $\oplus$ denotes the concatenation operation between two matrices. Then we use $\boldsymbol{c}_i^{'}$ and N_{e_i} to compute a structure based context representation of $\boldsymbol{\epsilon}_i = \sigma\left(\sum c_{ij}^{'} \boldsymbol{n}_{ij}^{'}\right)$, where $n_{ij} \in N_{e_i}$ and σ is Sigmoid function.

In order to capture various types of relations between e_i and its neighbors, we further perform multi-head attention on each entity, based on multiple linear transformation matrices. Finally, we get a structure based context representation $\tilde{\boldsymbol{e}}_i = [\boldsymbol{\epsilon}_i^0 \oplus ... \oplus \boldsymbol{\epsilon}_i^M]$, where $\boldsymbol{\epsilon}_i^m$ refers to the context representation obtained with the m-th head, and $\tilde{\boldsymbol{e}}_i$ is the concatenated representation based on the attention of all M heads.

Contextual Text Encoder Each entity e is also associated with a context sentence $[w_1, ..., w_l]$. To incorporate the local context information, we first apply a bi-directional long short-term memory (LSTM) (Graves and Schmidhuber, 2005) network to get the encoder hidden states $\boldsymbol{H}_s = [\boldsymbol{h}_1, ..., \boldsymbol{h}_l]$, where $\boldsymbol{h}_i$ represents the hidden state of w_i. Then we compute a bilinear attention weight for each word w_i: $\mu_i = \boldsymbol{e}^\top \boldsymbol{W}_s \boldsymbol{h}_i, \boldsymbol{\mu}^{'} = \text{Softmax}(\boldsymbol{\mu})$, where $\boldsymbol{W}_s$ is a bilinear term. We finally get the context representation $\hat{\boldsymbol{e}} = \boldsymbol{\mu}^{'\top} \boldsymbol{h}_i$.

Gated Combination To combine the graph-based representation $\tilde{\boldsymbol{e}}$ and local context based representations $\hat{\boldsymbol{e}}$, we design a gate function to balance these two types of information:

$$\boldsymbol{g}_e = \sigma(\tilde{\boldsymbol{g}}_e), \quad \boldsymbol{e} = \boldsymbol{g}_e \odot \tilde{\boldsymbol{e}} + (1 - \boldsymbol{g}_e) \odot \hat{\boldsymbol{e}}$$

where $\boldsymbol{g}_e$ is an entity-dependent gate function of which each element is in $[0, 1]$, $\tilde{\boldsymbol{g}}_e$ is a learnable parameter for each entity e, σ is a Sigmoid function, and $\odot$ is an element-wise multiplication.

Training and Prediction To optimize both entity and relation representations, following TransE (Bordes et al., 2013), we assume the relation between two entities can be interpreted as translations operated on the entity representations, namely $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ if (h, r, t) holds. Therefore, for each tuple (e_i^h, r_i, e_i^t), we can compute their distance score: $F(e_i^h, r_i, e_i^t) = \| \boldsymbol{e}_i^h + \boldsymbol{r}_i - \boldsymbol{e}_i^t \|_2^2$. We use marginal loss to train the model:

$$Loss = \sum_{(e_i^h, r_i, e_i^t) \in K} \sum_{(\bar{e}_i^h, \bar{r}_i, \bar{e}_i^t) \in \bar{K}} \max(0, \gamma + F(e_i^h, r_i, e_i^t) - F(\bar{e}_i^h, \bar{r}_i, \bar{e}_i^t))$$

where (e^h, r, t^h) is a positive tuple and $(\bar{e}^h, \bar{r}^h, \bar{t}^h)$ is a negative tuple, and γ is a margin. The negative tuples are generated by either replacing the head or the tail entity of positive tuples with a randomly chosen different entity.

Title	**Snail** transcription factor negatively regulates **maspin tumor suppressor** in **human prostate cancer** cells		
Entities	Related: **nasopharyngeal carcinoma**; diallyl disulfide		
Output	**Human (Neal et al., 2012)**	**System**	**Post-edited by Human**
Abstract	Background: **Maspin**, a putative **tumor suppressor** that is down-regulated in breast and **prostate cancer**, has been associated with decreased cell motility. **Snail** transcription factor is a zinc finger protein that is increased in breast cancer and is associated with increased tumor motility and invasion by induction of epithelial-mesenchymal transition (EMT). We investigated the molecular mechanisms by which **Snail** increases **tumor** motility and invasion utilizing **prostate cancer** cells. Methods: Expression levels were analyzed by **RT-PCR** and **western blot** analyses. Cell motility and invasion assays were performed , while **Snail** regulation and binding to **maspin** promoter was analyzed by luciferase reporter and chromatin immunoprecipitation (ChIP) assays. Results: **Snail** protein expression was higher in different **prostate cancer** cells lines as compared to normal prostate epithelial cells.	Background: **Snail** is a multifunctional protein that plays an important role in the pathogenesis of **prostate cancer**. *However*, it has been shown *to be* associated with poor prognosis. The purpose of this study *was* to investigate the effect of *negatively* on the expression of ***maspin*** in **human nasopharyngeal carcinoma** cell lines. Methods: *Quantitative real-time PCR* and western blot analysis were used to determine *whether the demethylating agent was investigated by quantitative* ***RT-PCR*** *(qRT-PCR) and* ***Western blotting***. Results showed that the binding protein plays a significant role in the regulation of **tumor** growth and progression.	Background: **Snail** is a multifunctional protein that plays an important role in the pathogenesis of **prostate cancer**. It has been shown associated with poor prognosis. The purpose of this study is to investigate the negative effect of on the expression of **Maspin** in **human nasopharyngeal carcinoma** cell lines. Methods: Quantitative **RT-PCR** *(qRT-PCR)* and **western blot analyses** were used to determine *correlation of the two proteins expressions.* Results showed that the binding protein plays a significant role in the regulation of **tumor** growth and progression.
Conclusion and Future work	Collectively, our results indicate for the first time that **Snail** can negatively regulate **maspin** through direct promoter repression resulting in increased migration and invasion in prostate **cancer** cells. This study reveals a novel mechanism of how **Snail** may function and show the importance of therapeutic targeting of **Snail** signaling in future.	In summary, our study demonstrates that **Snail** negatively *inhibited* the expression of **Maspin** in **human nasopharyngeal carcinoma** cell lines *and in vitro*. Our results indicate that *the combination of the demethylating agent* might be a potential therapeutic target for the treatment of **prostate cancer**.	In summary, our study *in vitro* demonstrates that **Snail** negatively *inhibits* the expression of **Maspin** in **human nasopharyngeal carcinoma** cell lines. Our results *further* indicate that ***Maspin*** might be a potential therapeutic target for the treatment of **prostate cancer**.
New Title	Role of **maspin** in cancer (Berardi et al., 2013)	The role of ***nasopharyngeal carcinoma*** in the rat model of ***prostate cancer*** cells	The role of ***Maspin*** in the rat model of ***nasopharyngeal carcinoma*** cells

Table 1: Comparison of Human and System Written Paper Elements (bold words are topically related entities; italic words show human edits)

After training, for each pair of indirectly connected entities e_i, e_j and a relation type r, we compute a score y to indicate the probability that (e_i, r, e_j) holds, and obtain an enriched knowledge graph $\widetilde{K} = [(e^h_{\kappa+1}, r_{\kappa+1}, e^t_{\kappa+1}, y_{\kappa+1})...]$.

2.4 New Paper Writing

In this section, we use title-to-abstract generation as a case study to describe the details of our paper writing approach. Other tasks (abstract-to-conclusion and future work, and conclusion and future work-to-title) follow the same architecture.

Given a reference title $\tau = [w_1, ..., w_l]$, we apply the knowledge extractor (Section 2.2) to extract entities from τ. For each entity, we retrieve a set of related entities from the enriched knowledge graph $\widetilde{K}$ after link prediction. We rank all the related entities by confidence scores and select up to 10 most related entities $E_\tau = [e^\tau_1, ..., e^\tau_v]$. Then we feed τ and E_τ together into the paper generation framework as shown in Figure 2. The framework is based on a hybrid approach of a Mem2seq model (Madotto et al., 2018) and a pointer generator (Gu et al., 2016; See et al., 2017). It allows us to balance three types of sources for each time step during decoding: the probability of generating a token from the entire word vocabulary based on language model, the probability of copying a word from the reference title, such as *regulates* in Table 1, and the probability of incorporating a related entity, such as *Snail* in Table 1. The output is a paragraph $Y = [y_1, ..., y_o]$.[3]

Reference Encoder For each word in the refer-

[3]During training, we truncate both of the input and the output to around 120 tokens to expedite training. We label the words with frequency < 5 as Out-of-vocabulary.

ence title, we randomly embed it into a vector and obtain $\boldsymbol{\tau} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_l]$. Then, we apply a bi-directional Gated Recurrent Unit (GRU) encoder (Cho et al., 2014) on $\boldsymbol{\tau}$ to produce the encoder hidden states $\boldsymbol{H} = [\boldsymbol{h}_1, ..., \boldsymbol{h}_l]$.

Decoder Hidden State Initialization Not all predicted entities are equally relevant to the title. For example, for the title in Table 2, we predict multiple related entities including *nasopharyngeal carcinoma* and *diallyl disulfide*, but *nasopharyngeal carcinoma* is more related because *nasopharyngeal carcinoma* is also a cancer related to *snail transcription factor*, while *diallyl disulfide* is less related because *diallyl disulfide*'s anticancer mechanism is not closely related to *maspin tumor suppressor*. We propose to apply memory-attention networks to further filter the irrelevant ones. Recent approaches (Sukhbaatar et al., 2015; Madotto et al., 2018) show that compared with soft-attention, memory-based multihop attention is able to refine the attention weight of each memory cell to the query multiple times, drawing better correlations. Therefore, we apply a multihop attention mechanism to generate the initial decoder hidden state.

Given the set of related entities $E = [e_1, ..., e_v]$, we randomly initialize their vector representation $\boldsymbol{E} = [\boldsymbol{e}_1, ..., \boldsymbol{e}_v]$ and store them in memories. Then we use the last hidden state of reference encoder $\boldsymbol{h}_l$ as the first query vector $\boldsymbol{q}_0$, and iteratively compute the attention distribution over all memories and update the query vector:

$$p_{ki} = \boldsymbol{\nu}_k^\top \tanh\left(\boldsymbol{W}_q^k \boldsymbol{q}_{k-1} + \boldsymbol{U}_e^k \boldsymbol{e}_i + \boldsymbol{b}_k\right)$$
$$\boldsymbol{q}_k = \boldsymbol{p}_k^\top \boldsymbol{e} + \boldsymbol{q}_{k-1}$$

where k denotes the k-th hop among φ hops in total.[4] After φ hops, we obtain $\boldsymbol{q}_\varphi$ and take it as the initial hidden state of the GRU decoder.

Memory Network To better capture the contribution of each entity e_j to each decoding output, at each decoding step i, we compute an attention weight for each entity and apply a memory network to refine the weights multiple times. We take the hidden state $\tilde{\boldsymbol{h}}_i$ as the initial query $\tilde{\boldsymbol{q}}_0 = \tilde{\boldsymbol{h}}_i$ and iteratively update it:

$$\tilde{p}_{kj} = \nu_k^\top \tanh\left(\widetilde{\boldsymbol{W}}_{\tilde{q}}^k \tilde{\boldsymbol{q}}_{k-1} + \widetilde{\boldsymbol{U}}_e^k \boldsymbol{e}_j + \boldsymbol{W}_{\hat{c}} \hat{c}_{ij} + b_k\right)$$
$$\boldsymbol{u}_{ik} = \tilde{\boldsymbol{p}}_k^{'\top} \boldsymbol{e}_j, \quad \tilde{\boldsymbol{q}}_k = \boldsymbol{u}_{ik} + \tilde{\boldsymbol{q}}_{k-1}$$

[4] We set $\varphi = 3$ since it performs the best on the development set.

where $\hat{\boldsymbol{c}}_{ij} = \sum_{m=0}^{i-1} \boldsymbol{\beta}_{mj}$ is an entity coverage vector and $\boldsymbol{\beta}_i$ is the attention distribution of last hop $\boldsymbol{\beta}_i = \tilde{\boldsymbol{p}}_\psi^{'}$, and ψ is the total number of hops. We then obtain a final memory based context vector for the set of related entities $\boldsymbol{\chi}_i = \boldsymbol{u}_{i\psi}$.

Reference Attention Our reference attention is similar to (Bahdanau et al., 2015; See et al., 2017), which aims to capture the contribution of each word in the reference title to the decoding output. At each time step i, the decoder receives the previous word embedding and generate decoder state $\tilde{\boldsymbol{h}}_i$, the attention weight of each reference token is computed as:

$$\alpha_{ij} = \boldsymbol{\varsigma}^\top \tanh\left(\boldsymbol{W}_h \tilde{\boldsymbol{h}}_i + \boldsymbol{W}_\tau \boldsymbol{h}_j + \boldsymbol{W}_{\tilde{c}} \tilde{\boldsymbol{c}}_{ij} + \boldsymbol{b}_\tau\right)$$
$$\boldsymbol{\alpha}_i^{'} = \mathrm{Softmax}\left(\boldsymbol{\alpha}_i\right); \quad \boldsymbol{\phi}_i = \boldsymbol{\alpha}_i^{'\top} \boldsymbol{h}_j$$

$\tilde{\boldsymbol{c}}_{ij} = \sum_{m=0}^{i-1} \alpha_{mj}$ is a reference coverage vector, which is the sum of attention distributions over all previous decoder time steps to reduce repetition (See et al., 2017). $\boldsymbol{\phi}_i$ is the reference context vector.

Generator For a particular word w, it may occur multiple times in the reference title or in multiple related entities. Therefore, at each decoding step i, for each word w, we aggregate its attention weights from the reference attention and memory attention distributions: $P_\tau^i = \sum_{m|w_m=w} \boldsymbol{\alpha}_{im}^{'}$ and $P_e^i = \sum_{m|w \in e_m} \boldsymbol{\beta}_{im}$ respectively. In addition, at each decoding step i, each word in the vocabulary may also be generated with a probability according to the language model. The probability is computed from the decoder state $\tilde{\boldsymbol{h}}_i$, the reference context vector $\boldsymbol{\phi}_i$, and the memory context vector $\boldsymbol{\chi}_i$: $P_{gen} = \mathrm{Softmax}(\boldsymbol{W}_{gen}[\tilde{\boldsymbol{h}}_i; \boldsymbol{\phi}_i; \boldsymbol{\chi}_i] + \boldsymbol{b}_{gen})$, where $\boldsymbol{W}_{gen}$ and b_{gen} are learnable parameters. To combine P_τ, P_e and P_{gen}, we compute a gate $\boldsymbol{g}_\tau$ as a soft switch between generating a word from the vocabulary and copying words from the reference title τ or the related entities E: $\boldsymbol{g}_p = \sigma(\boldsymbol{W}_p^\top \tilde{\boldsymbol{h}}_i + \boldsymbol{W}_z^\top \boldsymbol{z}_{i-1} + \boldsymbol{b}_p)$, where $\boldsymbol{z}_{i-1}$ is the embedding of the previous generated token at step $i-1$. $\boldsymbol{W}_p$, $\boldsymbol{W}_z$, and $\boldsymbol{b}_p$ are learnable parameters, and σ is a Sigmoid function. We also compute a gate $\tilde{\boldsymbol{g}}_p$ as a soft switch between copying words from reference text and the related entities: $\tilde{\boldsymbol{g}}_p = \sigma(\boldsymbol{W}_\phi^\top \boldsymbol{\phi}_i + \boldsymbol{W}_\chi^\top \boldsymbol{\chi}_i + \tilde{\boldsymbol{b}}_p)$, where $\boldsymbol{W}_\phi$, $\boldsymbol{W}_\chi$, and $\tilde{\boldsymbol{b}}_p$ are learnable parameters.

The final probability of generating a token z at decoding step i can be computed by:

$$P(z_i) = \boldsymbol{g}_p P_{gen} + (1 - \boldsymbol{g}_p)\left(\tilde{\boldsymbol{g}}_p P_\tau + (1 - \tilde{\boldsymbol{g}}_p) P_e\right)$$

Dataset	# papers			# avg entities in Title / paper	# avg predicted related entities / paper
	Title-to-Abstract	Abstract-to-Conclusion and Future work	Conclusion and Future work-to-Title		
Training	22,811	22,811	15,902	4.8	-
Development	2,095	2,095	2,095	5.6	6.1
Test	2,095	2,095	2,095	5.7	8.5

Table 2: Paper Writing Statistics

Model	Title-to-Abstract		Abstract-to-Conclusion and Future Work		Conclusion and Future Work-to-Title	
	Perplexity	METEOR	Perplexity	METEOR	Perplexity	METEOR
Seq2seq (Bahdanau et al., 2015)	19.6	9.1	44.4	8.6	49.7	6.0
Editing Network (Wang et al., 2018b)	18.8	9.2	30.5	8.7	55.7	5.5
Pointer Network (See et al., 2017)	146.7	8.5	74.0	8.1	47.1	6.6
Our Approach (-Repetition Removal)	13.4	12.4	24.9	**12.3**	31.8	7.4
Our Approach	**11.5**	**13.0**	**18.3**	11.2	**14.8**	**8.9**

Table 3: Automatic Evaluation on Paper Writing for Diagnostic Tasks (%). The Pointer Network can be viewed as removing memory network part from our approach without repetition removal.

The loss function, combined with the coverage loss (See et al., 2017) for both reference attention and memory distribution, is presented as:

$$Loss = \sum_i -\log P(z_i) + \lambda \sum_i (\min(\alpha_{ij}, \tilde{\boldsymbol{c}}_{ij}) + \min(\beta_{ij}, \hat{\boldsymbol{c}}_{ij}))$$

where $P(z_i)$ is the prediction probability of the ground truth token z_i, and λ is a hyperparameter.

Repetition Removal Similar to many other long text generation tasks (Suzuki and Nagata, 2017), repetition remains a major challenge (Foster and White, 2007; Xie, 2017). In fact, 11% sentences in human written abstracts include repeated entities, which may mislead the language model. Following the coverage mechanism proposed by (Tu et al., 2016; See et al., 2017), we use a coverage loss to avoid any entity in reference input text or related entity receiving attention multiple times. We further design a new and simple masking method to remove repetition during the test time. We apply beam search with beam size 4 to generate each output, if a word is not a stop word or punctuation and it is already generated in the previous context, we will not choose it again in the same output.

3 Experiment

3.1 Data

We collect biomedical papers from the PMC Open Access Subset.[5] To construct ground truth for new title prediction, if a human written paper *A* cites a paper *B*, we assume the title of *A* is generated from *B*'s conclusion and future work session. We construct background knowledge graphs from 1,687,060 papers which include 30,483 entities and 875,698 relations. Tables 2 shows the detailed data statistics. The hyperparameters of our model are presented in the Appendix.

3.2 Automatic Evaluation

Previous work (Liu et al., 2016; Li et al., 2016; Lowe et al., 2015) has proven it to be a major challenge to automatically evaluate long text generation. Following the story generation work (Fan et al., 2018), we use METEOR (Denkowski and Lavie, 2014) to measure the topic relevance towards given titles and use perplexity to further evaluate the quality of the language model. The perplexity scores of our model are based on the language model[6] learned on other PubMed papers (500,000 titles, 50,000 abstracts, 50,000 conclusions and future work) which are not used for training or testing in our experiment.[7] The results are shown in Table 3. We can see that our framework outperforms all previous approaches.

3.3 Turing Test

Similar to (Wang et al., 2018b), we conduct Turing tests by a biomedical expert (non-native speaker) and a non-expert (native speaker). Each human judge is asked to compare a system output and a human-authored string, and select the better one.

[5] ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_package/

[6] https://github.com/pytorch/examples/tree/master/word_language_model

[7] The perplexity scores of the language model are in the Appendix.

Task	Input		Output	Domain Expert	Non-expert
End-to-End	Human Title	Different	Abstract (1st)	10	**30**
		Same		**30**	16
	System Abstract	Different	Conclusion and Future work	**12**	0
		Same		8	8
	System Conclusion and Future work	Different	Title	**12**	2
		Same		12	**25**
	System Title	Different	Abstract (2nd)	**14**	4
Diagnostic	Human Abstract	Different	Conclusion and Future work	12	**14**
		Same		**24**	20
	Human Conclusion and Future work	Different	Title	8	**12**
		Same		2	**10**

Table 4: Turing Test Human Subject Passing Rates (%). Percentages show how often a human judge chooses our system's output over human's when it is mixed with a human-authored string. If the output strings (e.g., abstracts) are based on the same input string (e.g., title), the Input condition is marked "Same", otherwise "Different".

BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	TER
59.6	58.1	56.7	55.4	73.3	35.2

Table 5: Evaluation on Human Post-Editing(%)

Table 4 shows the results on 50 pairs in each setting. We can see that *PaperRobot* generated abstracts are chosen over human-written ones by the expert up to 30% times, conclusion and future work up to 24% times, and new titles up to 12% times. We don't observe the domain expert performs significantly better than the non-expert, because they tend to focus on different aspects - the expert focuses on content (entities, topics, etc.) while the non-expert focuses on the language.

3.4 Human Post-Editing

In order to measure the effectiveness of *PaperRobot* acting as a wring assistant, we randomly select 50 paper abstracts generated by the system during the first iteration and ask the domain expert to edit them until he thinks they are informative and coherent. The BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and TER (Snover et al., 2006) scores by comparing the abstracts before and after human editing are presented in Table 5. It took about 40 minutes for the expert to finish editing 50 abstracts. Table 1 includes the post-edited example. We can see that most edits are stylist changes.

3.5 Analysis and Discussions

To better justify the function of each component, we conduct ablation studies by removing memory networks, link prediction, and repetition removal respectively. The results are shown in Table 6. We can see that the approach without memory networks tends to diverge from the main topic, especially for generating long texts such as abstracts (the detailed length statistics are shown in Table 8). From Table 6 we can see the later parts of the abstract (Methods and Results) include topically irrelevant entities such as "*imipramine*" which is used to treat depression instead of human prostate cancer.

Link prediction successfully introduces new and topically related ideas, such as "*RT-PCR*" and "*western blot*" which are two methods for analyzing the expression level of Snail protein, as also mentioned in the human written abstract in Table 1. Table 7 shows more examples of entities which are related to the entities in input titles based on link prediction. We can see that the predicted entities are often genes or proteins which cause the disease mentioned in a given title, or other diseases from the same family.

Our simple beam search based masking method successfully removes some repeated words and phrases and thus produces more informative output. The plagiarism check in Table 9 shows our model is creative, because it's not simply copying from the human input.

3.6 Remaining Challenges

Our generation model is still largely dependent on language model and extracted facts, and thus it lacks of knowledge reasoning. It generates a few incorrect abbreviations such as *"Organophosphates(BA)"*, *"chronic kidney disease(UC)"* and *"Fibrosis(DC)"*) because they appear rarely in the training data and thus their contextual representations are not reliable. It also generates some incorrect numbers (e.g., *"The patients were divided into four groups :* ***Group 1*** *,* ***Group B****..."*) and pronouns (e.g., "*A* ***63-year-old man*** *was referred to our hospital ...* ***she*** *was treated with the use of the descending coronary artery"*).

Output	Without Memory Networks	Without Link Prediction	Without Repetition Removal
Abstract	Background: **Snail** has been reported to exhibit a variety of biological functions. In this study, we investigated the effect of negatively on **maspin** demethylation in **human prostate cancer** cells. Methods: Quantitative real-time PCR and western blot analysis were used to investigate the effects of the demethylating agent on the expression of the protein kinase (TF) gene promoter. Results: The results showed that the presence of a single dose of 50 μM in a dose-dependent manner, whereas the level of the BMP imipramine was significantly higher than that of the control group.	Background: **Snail** has been shown to be associated with poor prognosis. In this study, we investigated the effect of negatively on the expression of **maspin** in **human prostate cancer** cells. Methods: Cells were treated with a single dose of radiotherapy for 24 h, and was used to investigate the significance of a quantitative factor for the treatment of the disease. Results: The remaining controls showed a significant increase in the G2/M phase of the **tumor suppressor protein** ($p<0.05$).	Background: **Snail** is a major health problem in human malignancies. However, the role of **Snail** on the expression of **maspin** in **human prostate cancer** cells is not well understood. The aim of this study was to investigate the effect of **Snail** on the expression of **maspin** in **human prostate cancer** cells. Methods: The expression of the expression of **Snail** and **maspin** was investigated using quantitative RT-PCR and western blot analysis. Results: The remaining overall survival (OS) and overall survival (OS) were analyzed.
Conclusion and Future work	In summary, our study demonstrated that negatively inhibited the expression of the BMP imipramine in **human prostate cancer** cells. Our findings suggest that the inhibition of **maspin** may be a promising therapeutic strategy for the treatment.	In summary, our results demonstrate that negatively inhibited the expression of **maspin** in **human prostate cancer** cells. Our findings suggest that the combination of radiotherapy may be a potential therapeutic target for the treatment of disease.	In summary, our results demonstrate that **snail** inhibited the expression of **maspin** in **human prostatic** cells. The expression of **snail** in PC-3 cells by **snail**, and the expression of **maspin** was observed in the presence of the expression of **maspin**.
New Title	Protective effects of homolog on **human breast cancer** cells by inhibiting the Endoplasmic Reticulum Stress	The role of **prostate cancer** in **human breast cancer** cells	The role of **maspin** and **maspin** in human **breast cancer** cells

Table 6: Ablation Test Results on the Same Title in Table 1

Titles	**Predicted Related Entities**
Pseudoachondroplasia/COMP translating from the bench to the bedside	osteoarthritis; skeletal dysplasia; thrombospondin-5
Role of **ceramide** in **diabetes mellitus**: evidence and mechanisms	diabetes insulin ceramide; metabolic disease
Exuberant clinical picture of **Buschke-Fischer-Brauer palmoplantar keratoderma** in bedridden patient	neoplasms; retinoids; autosomal dominant disease
Relationship between **serum adipokine** levels and **radiographic progression** in patients with **ankylosing spondylitis**	leptin; rheumatic diseases; adiponectin; necrosis; DKK-1; IL-6-RFP

Table 7: More Link Prediction Examples (bold words are entities detected from titles)

	Abstract	**Conclusion and Future Work**	**Title**
System	112.4	88.1	16.5
Human	106.5	105.5	13.0

Table 8: The Average Number of Words of System and Human Output

Output	**1**	**2**	**3**	**4**	**5**
Abstracts	58.3	20.1	8.03	3.60	1.46
Conclusions	43.8	12.5	5.52	2.58	1.28
Titles	20.1	1.31	0.23	0.06	0.00

Table 9: Plagiarism Check: Percentage (%) of n-grams in human input which appear in system generated output for test data.

All of the system generated titles are declarative sentences while human generated titles are often more engaging (e.g., *"Does HPV play any role in the initiation or prognosis of endometrial adenocarcinomas?"*). Human generated titles often include more concrete and detailed ideas such as *"etumorType , An Algorithm of Discriminating Cancer Types for Circulating Tumor Cells or Cell-free DNAs in Blood"*, and even create new entity abbreviations such as *etumorType* in this example.

3.7 Requirements to Make PaperRobot Work: Case Study on NLP Domain

When a cool Natural Language Processing (NLP) system like *PaperRobot* is built, it's natural to ask whether she can benefit the NLP community itself. We re-build the system based on 23,594 NLP papers from the new ACL Anthology Network (Radev et al., 2013). For knowledge extraction we apply our previous system trained for the NLP domain (Luan et al., 2018). But the results are much less satisfactory compared to the

biomedical domain. Due to the small size of data, the language model is not able to effectively copy out-of-vocabulary words and thus the output is often too generic. For example, given a title "*Statistics based hybrid approach to Chinese base phrase identification*", *PaperRobot* generates a fluent but uninformative abstract "*This paper describes a novel approach to the task of Chinese-base-phrase identification. We first utilize the solid foundation for the Chinese parser, and we show that our tool can be easily extended to meet the needs of the sentence structure.*".

Moreover, compared to the biomedical domain, the types of entities and relations in the NLP domain are rather coarse-grained, which often leads to inaccurate prediction of related entities. For example, for an NLP paper title "*Extracting molecular binding relationships from biomedical text*", *PaperRobot* mistakenly extracts "*prolog*" as a related entity and generates an abstract "*In this paper, we present a novel approach to the problem of extracting relationships among the* ***prolog*** *program. We present a system that uses a macromolecular binding relationships to extract the relationships between the abstracts of the entry. The results show that the system is able to extract the most important concepts in the* ***prolog*** *program.*".

4 Related Work

Link Prediction. Translation-based approaches (Nickel et al., 2011; Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Ji et al., 2015a) have been widely exploited for link prediction. Compared with these studies, we are the first to incorporate multi-head graph attention (Sukhbaatar et al., 2015; Madotto et al., 2018; Veličković et al., 2018) to encourage the model to capture multi-aspect relevance among nodes. Similar to (Wang and Li, 2016; Xu et al., 2017), we enrich entity representation by combining the contextual sentences that include the target entity and its neighbors from the graph structure. This is the first work to incorporate new idea creation via link prediction into automatic paper writing.

Knowledge-driven Generation. Deep Neural Networks have been applied to generate natural language to describe structured knowledge bases (Duma and Klein, 2013; Konstas and Lapata, 2013; Flanigan et al., 2016; Hardy and Vlachos, 2018; Pourdamghani et al., 2016; Trisedya et al., 2018; Xu et al., 2018; Madotto et al., 2018; Nie et al., 2018), biographies based on attributes (Lebret et al., 2016; Chisholm et al., 2017; Liu et al., 2018; Sha et al., 2018; Kaffee et al., 2018; Wang et al., 2018a; Wiseman et al., 2018), and image/video captions based on background entities and events (Krishnamoorthy et al., 2013; Wu et al., 2018; Whitehead et al., 2018; Lu et al., 2018). To handle unknown words, we design an architecture similar to pointer-generator networks (See et al., 2017) and copy mechanism (Gu et al., 2016). Some interesting applications include generating abstracts based on titles for the natural language processing domain (Wang et al., 2018b), generating a poster (Qiang et al., 2016) or a science news blog title (Vadapalli et al., 2018) about a published paper. This is the first work on automatic writing of key paper elements for the biomedical domain, especially conclusion and future work, and follow-on paper titles.

5 Conclusions and Future Work

We build a *PaperRobot* who can predict related entities for an input title and write some key elements of a new paper (abstract, conclusion and future work) and predict a new title. Automatic evaluations and human Turing tests both demonstrate her promising performance. *PaperRobot* is merely an assistant to help scientists speed up scientific discovery and production. Conducting experiments is beyond her scope, and each of her current components still requires human intervention: constructed knowledge graphs cannot cover all technical details, predicted new links need to be verified, and paper drafts need further editing. In the future, we plan to develop techniques for extracting entities of more fine-grained entity types, and extend *PaperRobot* to write related work, predict authors, their affiliations and publication venues.

Acknowledgments

The knowledge extraction and prediction components were supported by the U.S. NSF No. 1741634 and Tencent AI Lab Rhino-Bird Gift Fund. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 5th International Conference on Learning Representations*.

Rossana Berardi, Francesca Morgese, Azzurra Onofri, Paola Mazzanti, Mirco Pistelli, Zelmira Ballatore, Agnese Savini, Mariagrazia De Lisa, Miriam Caramanti, Silvia Rinaldi, et al. 2013. Role of maspin in cancer. *Clinical and translational medicine*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Benjamin L King, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2016. The comparative toxicogenomics database: update 2017. *Nucleic acids research*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the 9th Workshop on Statistical Machine Translation*.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jacob G. Foster, Andrey Rzhetsky, and James A. Evans. 2015. Tradition and innovation in scientists research strategies. *American Sociological Review*.

Mary Ellen Foster and Michael White. 2007. Avoiding repetition in generated text. In *Proceedings of the 11th European Workshop on Natural Language Generation*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. In *Proceedings of the 2015 IEEE International Joint Conference on Neural Networks*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015a. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Ming Ji, Qi He, Jiawei Han, and Scott Spangler. 2015b. Mining strong relevance between heterogeneous entities from unstructured biomedical data. *Data Mining and Knowledge Discovery*, 29:976998.

Lucie-Aimée Kaffee, Hady Elsahar, Pavlos Vougiouklis, Christophe Gravier, Frederique Laforest, Jonathon Hare, and Elena Simperl. 2018. Learning to generate Wikipedia summaries for underserved languages from Wikidata. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research*.

Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of Text Summarization Branches Out*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Di Lu, Spencer Whitehead, Lifu Huang, Heng Ji, and Shih-Fu Chang. 2018. Entity-aware image caption generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Corey L. Neal, Veronica Henderson, Bethany N. Smith, Danielle McKeithen, Tisheeka Graham, Baohan T. Vo, and Valerie A. Odero-Marah. 2012. Snail transcription factor negatively regulates maspin tumor suppressor in human prostate cancer cells. *BMC Cancer*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*.

Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. 2018. Operation-guided neural networks for high fidelity data-to-text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Steven Pinker. 2014. Why academics stink at writing. *The Chronicle of Higher Education*.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from Abstract Meaning Representations. In *Proceedings of the 9th International Natural Language Generation conference*.

Yuting Qiang, Yanwei Fu, Yanwen Guo, Zhi-Hua Zhou, and Leonid Sigal. 2016. Learning to generate posters of scientific papers. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The acl anthology network corpus. *Language Resources and Evaluation*, pages 1–26.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Raghuram Vadapalli, Bakhtiyar Syed, Nishant Prabhu, Balaji Vasan Srinivasan, and Vasudeva Varma. 2018. When science journalism meets artificial intelligence: An interactive demonstration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Richard Van Noorden. 2014. Scientists may be reaching a peak in reading habits. *Nature*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *Proceedings of the 8th International Conference on Learning Representations*.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018a. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation*.

Qingyun Wang, Zhihao Zhou, Lifu Huang, Spencer Whitehead, Boliang Zhang, Heng Ji, and Kevin Knight. 2018b. Paper abstract writing through editing mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

Zhigang Wang and Juan-Zi Li. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*.

Spencer Whitehead, Heng Ji, Mohit Bansal, Shih-Fu Chang, and Clare Voss. 2018. Incorporating background knowledge into video description generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. 2018. Image captioning and visual question answering based on attributes and external knowledge. In *Proceedings of the 2018 IEEE transactions on pattern analysis and machine intelligence*.

Ziang Xie. 2017. Neural text generation: A practical guide. *arXiv preprint arXiv:1711.09534*.

Jiacheng Xu, Kan Chen, Xipeng Qiu, and Xuanjing Huang. 2017. Knowledge graph representation with jointly structural and textual encoding. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*.

Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, and Vadim Sheinin. 2018. SQL-to-text generation with graph-to-sequence model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.