The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Towards Topic-Aware Slide Generation For Academic Papers With Unsupervised Mutual Learning

Da-Wei Li, [*] [1] **Danqing Huang,** [2] **Tingting Ma,** [*] [3] **Chin-Yew Lin** [2]

[1] School of Software and Microelectronics, Peking University
[2] Microsoft Research Asia
[3] Harbin Institute of Technology
dwlee@pku.edu.cn, dahua@microsoft.com, hittingtingma@gmail.com, cyl@microsoft.com

Abstract

Slides are commonly used to present information and tell stories. In academic and research communities, slides are typically used to summarize findings in accepted papers for presentation in meetings and conferences. These slides for academic papers usually contain common and essential topics such as major contributions, model design, experiment details and future work. In this paper, we aim to automatically generate slides for academic papers. We first conducted an in-depth analysis of how humans create slides. We then mined frequently used slide topics. Given a topic, our approach extracts relevant sentences in the paper to provide the draft slides. Due to the lack of labeling data, we integrate prior knowledge of ground truth sentences into a log-linear model to create an initial pseudo-target distribution. Two sentence extractors are learned collaboratively and bootstrap the performance of each other. Evaluation results on a labeled test set show that our model can extract more relevant sentences than baseline methods. Human evaluation also shows slides generated by our model can serve as a good basis for preparing the final presentations.

Introduction

Slides are commonly used to present information and tell stories. In academic and research communities, slides are typically used to summarize findings in accepted papers for presentation in meetings and conferences. Automatically generating slides from papers aiming to reduce authors' time and efforts in slide creation would improve the authors' productivity. Slides for academic papers usually contain common and essential topics such as major contributions, model design, experiment details and future work. Given a paper, we aim to generate a set of draft slides covering the essential topics in academic presentations. In this way, we hope to offer paper authors a quick start slide creation experience rather than require them to create slide decks from scratch.

The task of generating slides for academic papers is very challenging and remains under-investigated. Early proposals (Masao and Kôiti 1999; Yasumura, Takeichi, and Nitta 2003; Sravanthi, Chowdary, and Kumar 2009) are mainly rule-based extractive methods with simple heuristics. They seldom report empirical evaluation results. Recently, some statistical learning approaches (Hu and Wan 2015; Wang, Wan, and Du 2017) have been proposed. They collect <paper, slide> pairs and use the alignments between slide contents and paper to learn the importance of phrases or sentences in the paper. They then design some heuristics to group sentences of high-importance as the final slides. We observe two major limitations in previous works. First, they do not consider slide topics during generation. The content in one slide should address one topic (e.g., paper contribution). Previous works extract sentences according to the sequential section order in a paper. The extracted sentences are grouped into slides based on predefined constraints without considering the topics of slides. Second, the slide datasets for training are small and not publicly available. Large and publicly available resources of academic paper slides are rare on the Web and difficult to collect. As far as we could find, the largest dataset (Hu and Wan 2015) only contains 1,000 pairs of slide and paper for training.

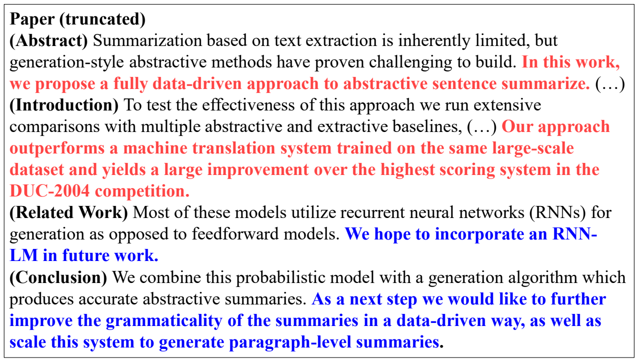

Paper (truncated)
(Abstract) Summarization based on text extraction is inherently limited, but generation-style abstractive methods have proven challenging to build. **In this work, we propose a fully data-driven approach to abstractive sentence summarize.** (…)
(Introduction) To test the effectiveness of this approach we run extensive comparisons with multiple abstractive and extractive baselines, (…) **Our approach outperforms a machine translation system trained on the same large-scale dataset and yields a large improvement over the highest scoring system in the DUC-2004 competition.**
(Related Work) Most of these models utilize recurrent neural networks (RNNs) for generation as opposed to feedforward models. **We hope to incorporate an RNN-LM in future work.**
(Conclusion) We combine this probabilistic model with a generation algorithm which produces accurate abstractive summaries. **As a next step we would like to further improve the grammaticality of the summaries in a data-driven way, as well as scale this system to generate paragraph-level summaries.**

Figure 1: An academic paper example. Sentences relevant to two topics Contribution (in red) and Future Work (in blue) are in different paper sections. Given the full paper, we retrieve relevant sentences for each topic to create the draft slides.

In this paper, we introduce a topic-aware paper to slide generation approach based on sentence selection. Take Figure 1 for example. Given the topic "Contribution" as a query, we extract relevant sentences (highlighted in red) in the paper

[*] This work is done while the authors were in internships in Microsoft Research Asia.

to provide the slide describing paper contribution. Similarly, we can generate slides concerning "Future Work" and other topics to obtain a presentation draft. In the next section, we will conduct a slide analysis and show how we mine the frequently used topics in academic presentations.

To deal with no available training data, we propose a simple and effective framework. We adapt mutual learning (Zhang et al. 2018) in the unsupervised setting with two sentence extractors in different views. Our first extractor is a neural-based model that aims to capture the distributional semantics of sentences. The second extractor is a log-linear model with predefined features. Due to the lack of labeling data, we integrate prior knowledge of both task-specific and general pre-trained model signal into the log-linear classifier to initiate the training. Specifically, we create a pseudo-target distribution using the log-linear model with heuristic weights, assuming sentences that meet more priors are more likely to be relevant. During mutual learning, two extractors learn collaboratively by teaching each other and bootstrap their performances.

We evaluate our approach on a labeled set containing 100 papers. Experimental results show that our method outperforms existing baselines by extracting more topic-relevant sentences. Human evaluation also confirms the quality of our generation outputs to serve as a presentation basis.

To summarize, our contributions include:

- We conduct an in-depth slide analysis, and have mined frequently covered topics into the slide generation process.
- We adapt mutual learning in the unsupervised setting, where we provide a general and flexible framework for integrating prior knowledge to initiate the training. Experiment results demonstrate that our approach is simple and effective.

Task Overview

We first conduct an analysis between papers and slides, then introduce our task formulation and test set.

Slide Analysis

Slide topics are important for understanding what to extract from papers. We aim to analyze the topics covered in academic presentations. By collecting and checking 50 presentations (1,127 slides in total) with their corresponding papers, we answer the following questions:

Q1 (Topic Popularity): What are the frequently covered topics in academic presentations? To categorize the topics, we calculate the frequency of slide titles and manually merge titles under the same topic. For example, slides titled "our contributions" or "this work" are categorized into the topic "Contribution". Table 1 shows the topics that are commonly seen across presentations.

Q2 (Extraction Coverage): For slides of each topic, how many textual contents are extracted from the corresponding papers? The feasibility of extraction-based systems depends on whether slide contents are extracted from their corresponding paper or external resources. To calculate the extraction coverage, we first create alignments from slide contents (bullet per unit) to paper sentences. To speed up the alignment process, we first retrieve the 5 most similar sentences for each slide bullet based on text similarity. Then we manually verify if the alignment is correct[1]. Overall, 85.16% of the slide bullets can be aligned to corresponding sentences in the paper, which confirms the feasibility of the extraction-based approaches. From Table 1, we can see that topics such as "Baseline" and "Future Work" have very high extraction coverage, while slide contents for "Task Background" are more likely to be generated from external resources with lower extraction rate.

Q3 (Extraction Distance): For all bullets in one slide, are the aligned sentences nearby in the paper? As we mentioned that previous works (Hu and Wan 2015; Wang, Wan, and Du 2017) extract sentences according to paper section orders, it is important to see if this assumption is valid that contents in the actual author-generated slides follow similar orders. Given a slide containing text bullets with each aligned to a paper sentence, we define the extraction distance as the average pairwise-distances of the aligned sentences:

$$avg(\sum_{k}\sum_{l=k+1}|pos_{a_k}-pos_{a_l}|) \qquad (1)$$

where a_k is the aligned paper sentence for the kth bullet in a slide, and $pos_i = \frac{i}{N}$ is ith sentence position in a paper with N sentences. Shorter extraction distance indicates contents within a slide are extracted from nearby sentences in a paper (e.g., within a paragraph or a section). From the 4th column in Table 1, we can see that some slide topics have larger extraction distances than others. For example, "Contribution" has large extraction distance, which indicates sentences for this topic might be extracted from different sections in the paper (e.g., abstract, introduction or conclusion). This observation further motivates our approach by retrieving topic-relevant sentences given slide topic as a query, instead of sequentially extracting sentences in the paper.

Slide Topic	**Popularity**	**Ext Cov. (#text bullets)**	**Ext Dist.**
Task Background	100%	70.23%(1,154)	0.074
Related Work	86%	82.14%(1,034)	0.158
Contribution	90%	87.14%(995)	0.693
Approach	100%	74.39%(70)	0.481
Dataset	84%	89.12%(310)	0.298
Baseline	88%	90.17%(255)	0.121
Result	100%	72.33%(101)	0.248
Conclusion	76%	76.25%(258)	0.799
Future Work	72%	93.68%(186)	0.011

Table 1: Statistics of our slide topic analysis. Topics with high popularity and extraction coverage are ideal to generate the slide draft via extractive approach.

[1] We only consider slide bullets with more than 3 words.

Task Formulation and Dataset

Based on the above observations, in this paper we view the slide generation task as sentence selection given slide topics as the query. Given a slide topic T and a paper P with N sentences $\{S_1, S_2, ..., S_N\}$, the goal is to select topic-relevant sentences.

We start with slide topics with both high popularity (Q1) and extraction coverage (Q2). According to the statistics shown in Table 1, we select four topics for experiments: $\{$Contribution, Dataset, Baseline, Future Work$\}$. For other topics such as "Approach" and "Result", we plan to generate the related slides with rule-based approach, since they contain mostly equations, figures or tables that are not in our current consideration.

Since datasets of papers and slides in previous works are not publicly available, we use the ACL Anthology Reference Corpus (Bird et al. 2008) as the unlabeled corpus of papers for our unsupervised learning.

ACL Anthology Reference Corpus: It contains 22,878 publications in the ACL Anthology up to December 2015. The corpus provides the original PDF format as well as the automatically extracted text with logical structure (e.g., section information) via ParsCit (Councill, Giles, and Kan 2008).

Testset Annotation: We evaluate our approach via (1) performance of relevant sentence selection from papers; (2) comparison with human-generated slides. We select 100 papers from the ACL corpus, which we found corresponding presentations on the Web. Two annotators with rich research experience are asked to annotate relevant sentences in the papers for each of the four slide topics[2]. All topics achieve high inter-annotator agreements (Kappa > 0.85) regarding relevance of sentences. We use the union sets of their annotations as the final ground truth sentences for evaluation. On average, there are 3.54 relevant sentences per paper per topic.

Approach

Input all sentences in a paper, our goal is to select sentences that are relevant to a slide topic. Our learning paradigm is based on mutual learning with two models updating collaboratively. In this section, we will first introduce the two models and then describe the learning algorithm. Overview of our approach is shown Figure 2.

Neural Sentence Selection Model

Our neural sentence selection model is based on the work of Zhou et al. (2018), coupled with a hierarchical document encoder and a sentence selector of pointer network (Vinyals, Fortunato, and Jaitly 2015).

Paper Encoder: It encodes the sentences in two levels, namely sentence level and document level. Given a paper $P = (S_1, S_2, ..., S_N)$ with N sentences, the sentence-level encoder first constructs a basic representation $\tilde{s_j}$ for each sentence S_j containing words $(x^j_{n_1}, x^j_{n_2}, ..., x^j_{n_j})$. It is implemented as a single-layer biRNN with GRU (Cho et al. 2014). It reads words one-by-one from the sentence, producing a sequence of hidden states:

$$\overrightarrow{h}^j_i = \text{GRU}(\phi^{in}(x^j_i), \overrightarrow{h}^j_{i-1}), \tag{2}$$

$$\overleftarrow{h}^j_i = \text{GRU}(\phi^{in}(x^j_i), \overleftarrow{h}^j_{i+1}), \tag{3}$$

where ϕ^{in} maps each input word x^j_i to a fixed-dimensional vector. We concatenate the last forward and first backward GRU hidden states to get the basic sentence representation $\tilde{s_j} = [\overrightarrow{h}^j_{n_j}, \overleftarrow{h}^j_1]$. In the document level, we use another biRNN to read sentences in a paper. It then produces hidden states:

$$\overrightarrow{s}_j = \text{GRU}(\tilde{s_j}, \overrightarrow{s}_{j-1}), \tag{4}$$

$$\overleftarrow{s}_j = \text{GRU}(\tilde{s_j}, \overleftarrow{s}_{j+1}), \tag{5}$$

and we can get the final sentence representation $\boldsymbol{s_j} = [\overrightarrow{s}_j, \overleftarrow{s}_j]$.

Sentence Selector: At each decoding time step t, the selector receives the hidden state q_{t-1} of the previous selected sentence. The probability of sentence S_i being selected is calculated as:

$$p(S_i|\theta) = \text{softmax}(W_h \cdot \tanh(W_q q_t + W_s \boldsymbol{s_i})) \tag{6}$$

$$q_t = \text{GRU}(\boldsymbol{s_{t-1}}, q_{t-1}) \tag{7}$$

$$q_0 = \tanh(W_m \boldsymbol{s}_1 + b_m) \tag{8}$$

where W_h, W_q, W_s, W_m, b_m are learnable, and θ is the model parameters.

Log-Linear Classifier with Prior Knowledge

While on one hand the neural sentence selection model captures sentence semantics, on the other hand there are useful priors that give hints of ground truth sentences. Especially in the unsupervised setting, the integration of prior knowledge is key for the model to learn in good direction.

Inspired by the knowledge integration approaches in neural machine translation (Zhang et al. 2017; Ren et al. 2019), we encode the prior knowledge as features within a log-linear model. The sentence selection distribution is calculated as:

$$q(S_i|\gamma) = \frac{\exp(\gamma \cdot \phi(S_i, T))}{\sum_k \exp(\gamma \cdot \phi(S_k, T))} \tag{9}$$

where $\phi(S_i, T)$ is the feature vector of sentence S_i given topic T, and γ is the feature weights.

From the slide analysis in the previous section and the features used in previous works, we adopt (1) task-specific features; (2) general signals from existing pre-trained model as follows:

- **Keywords.** For sentences relevant to each slide topic, we have observed some textual patterns. We design sets of keywords for each topic. If a sentence contains a keyword, this feature value is set to 1, otherwise 0.
- **Belonging Section.** Sentences of some slide topics tend to appear in certain paper sections. For example, "Dataset" are always in the experiment-related sections. Similar to keyword feature, we design sets of section keywords for each topic. If a sentence's belonging section contains a section keyword, the feature value is set to 1, otherwise 0.

[2] Our annotation and code can be found at https://github.com/daviddwlee84/TopicAwarePaperSlideGeneration

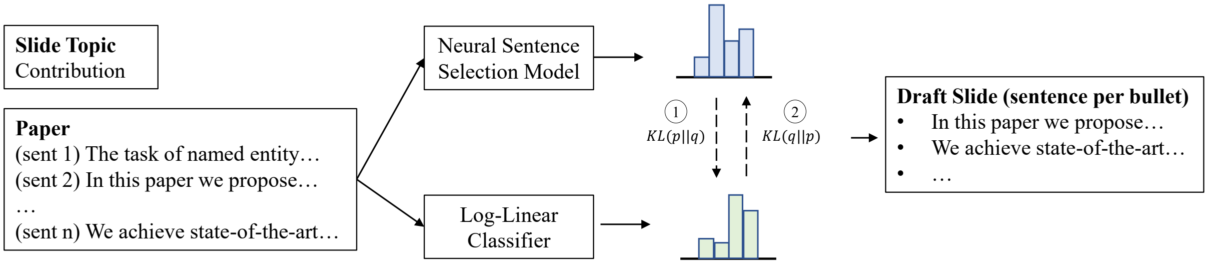

Slide Topic
Contribution
Paper
(sent 1) The task of named entity…
(sent 2) In this paper we propose…
…
(sent n) We achieve state-of-the-art…
Neural Sentence Selection Model
Log-Linear Classifier
① $KL(p||q)$
② $KL(q||p)$
Draft Slide (sentence per bullet)
• In this paper we propose…
• We achieve state-of-the-art…
• …

Figure 2: Overview of our topic-aware learning approach.

- **Sentence Position.** Similar to Hu and Wan (2015); Wang, Wan, and Du (2017), we argue that the sentence position in a paper is also important. For example, sentences related to "Future Work" are more likely to appear at the end of the paper. We use the normalized position $pos_i = \frac{i}{N}$ for sentence i as the feature value.
- **BERT-QA Signal.** To leverage the knowledge learned from the recent large pre-trained model, we incorporate the signal from the current state-of-the-art BERT-QA model (Devlin et al. 2019) (fine-tuned on SQUAD (Rajpurkar et al. 2016)) as a feature. We convert the sentence selection task to a QA problem for BERT. Given a slide topic such as "Contribution", we input the question "*what are the **contributions** in this paper?*" and retrieve the text span predicted by BERT. The feature value is set to 1 for the sentence containing the BERT output span.

More feature implementation details are described in the Supplementary.

Learning Algorithm

For each topic, we train two extractors iteratively to retrieve topic-relevant sentences in the paper. Our learning paradigm is inspired by mutual learning (Zhang et al. 2018), where several networks are trained collaboratively by teaching each other to bootstrap the performance in supervised settings.

In our case, without labeling data, the pseudo target distribution and training data are critical to initiate the training. As shown in Algorithm 1, our training consists of two stages. In the first stage, we create a seed subset from the large unlabeled corpus. We keep the papers containing topic-specific keywords (mentioned in subsection), which are assumed to have ground-truth topic-relevant sentences. For target distribution, we assign heuristic weights to the log-linear model and use its output as pseudo ground truth to guide the training. This is based on the assumption that sentences, which meet more priors, are more likely to be relevant. We train the two classifiers (i.e., neural-based model and log-linear model) alternately. For the neural sentence selection model, we minimize the KL loss function:

$$\mathcal{L}_{neural} = D_{KL}(p(S_i|\theta) \mid\mid q(S_i|\gamma)) \tag{10}$$

For the log-linear model, the KL loss function is:

$$\mathcal{L}_{log\_linear} = D_{KL}(q(S_i|\gamma) \mid\mid p(S_i|\theta)) \tag{11}$$

Two classifiers are updated iteratively until converge to each other.

In the second stage, we try to exploit more unlabeled data to augment the training. In each epoch, we randomly sample a batch of unlabeled papers as extra training data. The target distribution is updated with the model predictions.

Experiment

Evaluation Metrics

We evaluate the system performance using three metrics. Given a topic, we calculate: 1) accuracy: the percentage of test cases that have at least one correct sentence retrieved; 2) sentence-level classification precision/recall; 3) BLEU (Papineni et al. 2002): a widely used evaluation method in machine translation and text generation, to evaluate the content overlap between model outputs and reference sentences in the paper.

Baselines

For slide generation from academic papers, the most recent systems (Hu and Wan 2015; Wang, Wan, and Du 2017) are not publicly available for comparison, and they require some feature engineering which is difficult for re-implementation. Similar to these systems, we consider several publicly available baselines related to summarization:

- **LexRank** (Erkan and Radev 2004): A graph-based method that computes sentence importance based on the eigenvector centrality in a graph. The weight of edges represents the cosine similarity between sentences. We limit the number of its extracted sentences to 15% of the paper text length, as done by Wang, Wan, and Du (2017).
- **SumBasic** (Nenkova and Vanderwende 2005): A frequency-based summarizer that seamlessly integrates content selection and re-ranks depending on the previous choice of content by updating the probabilities of words. We limit the number of its extracted sentences similar to the LexRank baseline.
- **RSA-TFIDF, RSA-word2vec** (Baumel, Eyal, and Elhadad 2018): The state-of-the-art query-focused abstractive summarizer which injects query-sentence relevance into the pre-trained seq2seq summarization model (See, Liu, and Manning 2017). We input the topic word as query and output maximum 250 words as in its original setting.
- **BERT-QA**: We also compare with a QA baseline. Given an input of a paper and a question for each slide topic, we output the sentence where BERT-QA predicted text span is

in. This is the same as the feature we used in the log-linear classifier described in previous section.

All sentences in the paper are used as input for all models. Since the first two traditional extractive baselines are not query-focused, their outputs are assumed not to be specifically related to the target slide topic. Therefore, we further train two enhanced baselines (+selected selections) which only take sentences in specific sections as constrained input. For each slide topic, we select the sections using the "Belong Sections" feature in the log-linear classifier.

Implementations

Model Parameters. Our vocabulary size is set to 1,000,000 most frequent words. Hidden size of word embedding, sentence-level encoder, and document-level encoder in neural sentence selection model are set to 50, 256, and 256, respectively. For the log-linear classifier, we assigned weights γ with 0 on the position feature and 1 on the other features. We show the parameters of baselines in the Supplementary File.

Model Training. Since there are a few other documents besides academic papers in the corpus (e.g., volume catelogs), we apply a heuristic rule to only keep those with larger than 50 and less than 500 sentences as academic papers in the training. We initialize the model parameters randomly using a Gaussian distribution with Xavier scheme (Glorot and Bengio 2010). The word embedding matrix was initialized using pre-trained 50-dimension GloVe vectors (Pennington, Socher, and Manning 2014). We use Adam (Kingma and Ba 2015) as our optimizing algorithm. The learning rate for Adam optimizer α is set to 0.001. We use dropout (Srivastava et al. 2014) as regularization with probability p = 0.3 after the sentence level encoder and p = 0.2 after the document level encoder. The training process stops when the loss of two classifiers converges. Maximum training epochs are set to 20.

Model Inference. We use decode step $= 1$ and retrieve the top $K = 1, 3$ sentences with highest probabilities as outputs.

Result Analysis

Overall results are shown in Table 2. We report the results of both our neural-based model and log-linear model. As we can see, the two models perform comparably. In most topics, our models perform better than baseline approaches.

Baseline Performances. LexRank achieves a high accuracy of 0.806 in topic "Contribution", but performs much worse on other topics. We observe that it prefers to extract front sentences of the papers (mainly in the abstract and introduction sections), where contents related to "Contribution" are more likely to appear. Even with the heuristically selected section as constrained input, the two traditional extractive baselines, LexRank and SumBasic, do not improve too much. The query-focused baselines RSA-TFIDF and RSA-word2vec do not perform as well as in the DUC 2005 (Dang 2005), since there might exist some gaps of query types and document domains. BERT-QA is a strong baseline. It demonstrates the rich knowledge embodied in BERT which is pre-trained with large-scale corpus.

Algorithm 1 Training paradigm based on mutual learning

```
Require: Unlabeled papers $U$, neural-based model $C_1$, log-linear model $C_2$
Create seed subset $U'$ with keyword filtering
Initialize pseudo-target distribution with log-linear model weights $\gamma$;
while $\mathcal{L}_{neural}$ not converges do
    Update $C_1$ with $\mathcal{L}_{neural}$, $U'$;
    Update $C_2$ with $\mathcal{L}_{log\_linear}$, $U'$;
end while
for timestep $t = 1, \cdots, T_1$ do
    Sample a batch of unlabeled papers $U_t$
    Update $U' = U' + U_t$;
    Use $C_1$ to label $U'$;
    Update $C_2$ with pseudo-labeled $U'$;
    Use $C_2$ to label $U'$;
    Update $C_1$ with pseudo-labeled $U'$;
end for
```

Performance of different topics. Among the topics, our model performs the best in "Contribution" and "Future Work", as it retrieves relevant sentences for over 86.7% and 71.8% of papers in the test set, respectively. Both of our models (i.e., neural-based and log-linear models) obtain the best scores on precision and BLEU. For our models, as well as other systems, topic "Baseline" is the most challenging. Approximately, only 20% of test cases have been solved by retrieving sentences describing baselines. Through our analysis, we observe that sentences describing "Baseline" are usually short, containing only model names and references. Our models might have wrongly learned the textual pattern, due to factors such as sentences containing reference (e.g., "*Reinforcement learning has been widely used in NLP **(Liang, 2005)**.*").

Effect of Prior Knowledge and Mutual Learning. To check the of validity of our prior knowledge, we use two features (Belong Section and BERT-QA) as end-to-end filtering hard rules to directly retrieve relevant sentences. For example, using Belonging Section feature, we retrieve all sentences in the sections of which section name contains the keyword for "Contribution" as output. Results are shown in Table 2, "Rule@section " and "BERT QA" respectively. We can see that both features have considerable performances, which demonstrates the effectiveness of prior knowledge. Additionally with mutual learning, our models further bootstrap the performance with a large margin. This indicates that mutual learning can well utilize the prior knowledge and contribute to the performance gain.

Performance Curve

To better understand our learning paradigm, we plot the accuracy curves of our neural sentence selection model in Figure 3. For topic "Baseline", we can see a steady performance improvement during mutual learning. As for "Contribution", our model already obtains a relatively high accuracy in the first epoch, since pseudo labels provided by BERT-QA and keyword priors are representative of the ground truth.

When we start to exploit more unlabeled data (epoch 10),

Topic	*Contribution*			*Dataset*			*Baseline*			*Future Work*		
	acc	p / r	bleu	acc	p / r	bleu	acc	p / r	bleu	acc	p / r	bleu
LexRank	0.81	0.06/0.39	0.06	0.22	0.10/0.11	0.02	0.37	0.02/0.19	0.03	0.21	0.01/0.12	0.01
LexRank*	0.62	0.18/0.21	0.22	0.09	0.04/0.05	0.05	0.23	0.04/0.11	0.05	0.10	0.03/0.05	0.04
SumBasic	0.46	0.02/0.12	0.05	0.25	0.01/0.10	0.02	0.24	0.01/0.10	0.02	0.14	0.01/0.08	0.01
SumBasic*	0.36	0.10/0.08	0.15	0.07	0.03/0.03	0.05	0.19	0.03/0.08	0.04	0.03	0.01/0.02	0.02
RSA-TFIDF	-	-	0.14	-	-	0.15	-	-	0.07	-	-	0.08
RSA-word2vec	-	-	0.16	-	-	0.10	-	-	0.07	-	-	0.09
BERT QA (sent1)	0.46	0.50/0.09	0.61	0.30	0.31/0.11	0.31	0.19	0.21/0.06	0.18	0.38	0.42/0.21	0.41
BERT QA (sent3)	0.63	0.38/0.19	0.40	**0.48**	0.21/0.21	0.218	0.31	0.11/0.10	0.10	0.55	0.24/0.32	0.21
Rule@section	**0.95**	0.13/**0.74**	0.13	0.35	0.05/**0.28**	0.08	**0.41**	0.02/**0.47**	0.05	0.38	0.03/0.32	0.04
Log-linear (sent1)	0.79	0.79/0.15	0.84	0.33	**0.33**/0.12	**0.32**	0.20	0.20/0.07	0.18	0.62	0.62/0.33	0.64
Log-linear (sent3)	0.86	0.77/0.21	0.77	0.40	0.27/0.14	0.28	0.26	0.18/0.10	0.16	0.63	0.58/0.34	0.60
Neural (sent1)	0.80	**0.80**/0.15	**0.85**	0.33	**0.33**/0.12	0.31	0.21	**0.21**/0.07	**0.21**	0.634	**0.63**/0.34	**0.66**
Neural (sent3)	0.87	0.71/0.22	0.71	0.44	0.28/0.18	0.25	0.31	0.16/**0.12**	0.16	**0.72**	0.50/**0.39**	0.58

Table 2: Overall performance on the 4 selected slide topics. “Neural” means our neural sentence selection model. “Log-linear” means our log-linear classifier. “-” indicates the system is not applicable on the metric. “*” indicates the input as selected sections.

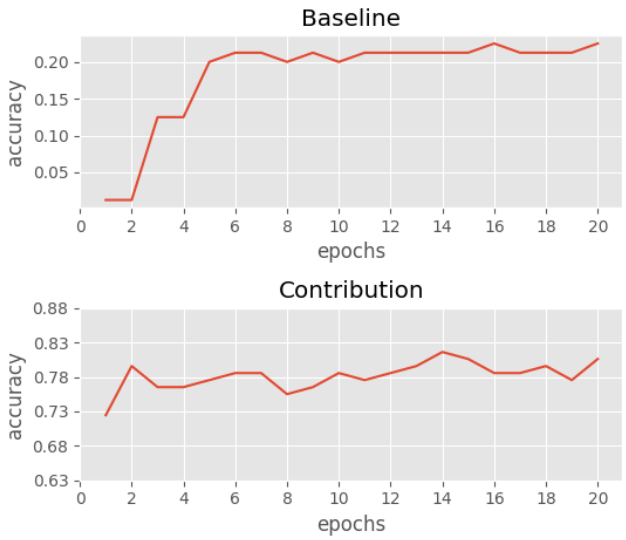


Figure 3: Accuracy curve of our neural-based model for topic “Baseline” and “Contribution” on test set.

our model learns at a stable pace and achieves the best performance in epoch 14. From the two figures, we can see our model’s robustness to the noise of increasing unlabeled data.

Discussion on Initialization

In this subsection, we delve into the initialization of the log-linear classifier, which is crucial to our unsupervised approach. Since we encode priors in the log-linear classifier to create pseudo target distribution, a reasonable initialization will provide a good training direction, and vice versa. We show the ablation results using different log-linear feature weight initialization in Table 3. From the table, we find that all three priors encoded in the log-linear model are useful, especially keywords and BERT-QA.

Features	acc	p / r	bleu
All	**0.63**	**0.63 / 0.34**	**0.66**
-keyword only	0.58	0.58 / 0.31	0.61
-BERT-QA only	0.56	0.56 / 0.30	0.61
-section only	0.16	0.16 / 0.08	0.19

Table 3: Results of our neural-based model using different initialization feature weights of log-linear classifier for topic “Future Work”.

It shows the possibility to extend our approach to other domains utilizing both task-specific feature and general pretrained model signal. Using only section feature results in a huge performance drop on all metrics, which stress the importance of prior designs for initialization.

Human Evaluation

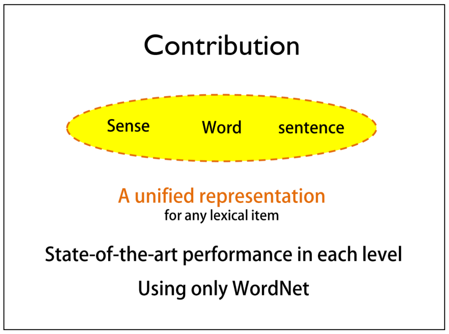
To further verify the quality of our generation outputs, we conduct the following human evaluation. For each of the four slide topics, we randomly choose 10 relevant slides generated by original authors from the test set. This sums up to total 40 evaluation cases. We pair up the human generated slides with our neural-based model’s output slides. Three students with rich experiences in academic research were asked to rate the slides from 1 (low) to 5 (high) in three aspects: (1) relevance: how relevant are the slides describing the given topic; (2) coverage: how many topic-relevant contents in papers have been covered by the slides; (3) overall: how well do the slides by our model serve as a basis for preparing the final presentations. From the evaluation results shown in Table 5, we can see that our model outputs are rated as less relevant than author-generated slides, since not all sentences retrieved by our model are topic-relevant. In terms of coverage, our model outputs (average score 3.52) are comparable to the original author-generated slides (average score 3.99). And the overall rating of our model is 3.54 (above average score

Ground truth	• The contributions of this paper are threefold. • First, we propose **a new unified representation of the meaning of an arbitrarily-sized piece of text**, referred to as a lexical item, using a sense-based probability distribution. • Second, we propose a novel alignment-based method for word sense disambiguation during semantic comparison. • Third, we demonstrate that this single representation can **achieve state-of-the-art performance** on three tasks...
Our model output	• The contributions of this paper are threefold. • First, we propose a new unified representation of the meaning of an arbitrarily-sized piece of text, referred to ... • Second, we propose a novel alignment-based method for word sense disambiguation during semantic...

Table 4: A case study for "Contribution". Words in bold are covered by author-generated slide in Figure 4.

	Relevance	Coverage	Overall
Author	4.33	3.99	-
Ours	3.56	3.52	3.54

Table 5: Human evaluation results on slides generated by original authors and our model.



Contribution

Sense Word sentence

A unified representation

for any lexical item

State-of-the-art performance in each level

Using only WordNet

Figure 4: Author-created slide of the example in Table 4.

3), which shows its potential to serve as draft slides.

The example in Table 4 shows that our model correctly retrieves 3 out of 4 ground truth sentences. More examples are shown in the Supplementary. Meanwhile, we observe two future directions to improve our slide generation approach: 1) **simplicity**; the author-generated slide (Figure 4) contains more abstract phrases and diagrams than our sentence-level outputs. 2) **slide structure**; currently we treat each retrieved sentence as a bullet in the slide without considering any hierarchical relations.

Related Work

Slide Generation for Academic Paper. Meshram and Phalke (2015) summarizes a review of slide generation approaches before 2015. Previous works are mostly extraction-based by scoring sentences in the paper. Yasumura, Takeichi, and Nitta (2003) calculates tf-idf term weights in the paper. Sentences with higher weights are selected for each section as form the final slides. PPSGen (Hu and Wan 2015) assumes sentences that contain more similar contents in the corresponding slides have higher scores. It trains an SVR model (Vapnik 1998) on 1,000 training paper and slide pairs to learn sentence importance in a paper. Then it selects sentences that maximize predefined objectives with constraints using Integer Linear Programming. Similarly, Wang, Wan, and Du (2017) learns phrase-level importance with 100 training pairs and optimizes heuristic objectives to generate slides with hierarchical relationship. The above approaches generate presentations in sequential order with the paper sections, without explicitly considering topics in the slide.

There exists other works that consider query-specific slide generation. One of the earliest slide generation approaches (Masao and Kôiti 1999) detects topics in a document based on word frequencies and semantic dependencies. For each topic, it retrieves text-similar sentences to generate the slides. SlidesGen (Sravanthi, Chowdary, and Kumar 2009) selects key phrases in the model and experiment sections as topic inputs to a query-specific summarizer QueSTS (Sravanthi, Chowdary, and Kumar 2008). QueSTS constructs a text graph where an edge exists between two sentence nodes if they are similar. For each query, it searches the graph for relevant sentences. However, no empirical results are reported in most early publications.

Query-Focused Summarization. Topic-aware slide generation can be also viewed as query-focused summarization task. This task was first introduced in DUC 2005 (Dang 2005). Current state-of-the-art methods are mainly unsupervised. Feigenblat et al. (2017) designs six query-related features and uses the Cross Entropy method to learn a global sentence selection policy. Later work (Erera et al. 2019) builds a section-based summarization system for academic papers using this method. Litvak and Vanetik (2017) uses Minimum Description Length (MDL) principle to select query-related frequent word sets. Baumel, Eyal, and Elhadad (2018) introduces an abstractive approach to consider query relevance into a pre-trained summarizer.

Conclusion

In this paper, we present a topic-aware paper to slide generation approach. With an in-depth analysis, we have mined frequently used slide topics. We design two sentence extractors and adapt mutual learning to update two models collaboratively. Due to the lack of labeling data, we integrate priors into the log-linear model to create pseudo-target distribution for initialization. Experiment results show that our model consistently outperforms baselines. Human evaluation also indicates our generated slides provide a good basis for preparing the final presentation. In the future, we plan to generate abstractive slides and explore more signals for training.

References

Baumel, T.; Eyal, M.; and Elhadad, M. 2018. Query Focused Abstractive Summarization: Incorporating Query Relevance, Multi-Document Coverage, and Summary Length Constraints into seq2seq Models. *CoRR* abs/1801.07704.

Bird, S.; Dale, R.; Dorr, B.; Gibson, B.; Joseph, M.; Kan, M.-Y.; Lee, D.; Powley, B.; Radev, D.; and Tan, Y. F. 2008. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation.*

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 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*, 1724–1734.

Councill, I.; Giles, C. L.; and Kan, M.-Y. 2008. ParsCit: an Open-source CRF Reference String Parsing Package. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08).*

Dang, H. T. 2005. DUC 2005: Evaluation of Question-Focused Summarization Systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, 48–55.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Erera, S.; Shmueli-Scheuer, M.; Feigenblat, G.; Peled Nakash, O.; Boni, O.; Roitman, H.; Cohen, D.; Weiner, B.; Mass, Y.; Rivlin, O.; Lev, G.; Jerbi, A.; Herzig, J.; Hou, Y.; Jochim, C.; Gleize, M.; Bonin, F.; Bonin, F.; and Konopnicki, D. 2019. A Summarization System for Scientific Documents. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing: System Demonstrations*, 211–216.

Erkan, G.; and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22: 457–479.

Feigenblat, G.; Roitman, H.; Boni, O.; and Konopnicki, D. 2017. Unsupervised Query-Focused Multi-Document Summarization Using the Cross Entropy Method. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 961–964.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

Hu, Y.; and Wan, X. 2015. PPSGen: Learning-Based Presentation Slides Generation for Academic Papers. *IEEE Transactions on Knowledge and Data Engineering* 27: 1085–1097.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of 3rd International Conference for Learning Representations*.

Litvak, M.; and Vanetik, N. 2017. Query-based summarization using MDL principle. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, 22–31.

Masao, U.; and Kôiti, H. 1999. Automatic Slide Presentation from Semantically Annotated Documents. In *Proceedings of the Workshop on Coreference and its Applications. Association for Computational Linguistics*.

Meshram, E. G.; and Phalke, M. D. 2015. Survey on Presentation Slides Generation for Academic Papers. *International Journal of Engineering Research* 5: 467–468.

Nenkova, A.; and Vanderwende, L. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005* 101.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

Ren, S.; Zhang, Z.; Liu, S.; Zhou, M.; and Ma, S. 2019. Unsupervised Neural Machine Translation with SMT as Posterior Regularization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 241–248.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1073–1083.

Sravanthi, M.; Chowdary, C. R.; and Kumar, P. S. 2008. QueSTS: A Query Specific Text Summarization System. In *FLAIRS Conference*, 219–223.

Sravanthi, M.; Chowdary, C. R.; and Kumar, P. S. 2009. SlidesGen: Automatic Generation of Presentation Slides for a Technical Paper Using Summarization. In *Proceedings of the Twenty-Second International FLAIRS Conference*, 284–289.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1): 1929–1958.

Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*, 2692–2700.

Wang, S.; Wan, X.; and Du, S. 2017. Phrase-Based Presentation Slides Generation for Academic Papers. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 196–202.

Yasumura, Y.; Takeichi, M.; and Nitta, K. 2003. A support system for making presentation slides. *Transactions of the Japanese Society for Artificial Intelligence* 18(4): 212–220.

Zhang, J.; Liu, Y.; Luan, H.; Xu, J.; and Sun, M. 2017. Prior Knowledge Integration for Neural Machine Translation using Posterior Regularization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1514–1523.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep Mutual Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4320–4328.

Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; and Zhao, T. 2018. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 654–663. Association for Computational Linguistics.