Automatic Webpage Briefing

Yimeng Dai[†], Rui Zhang[‡*], Jianzhong Qi[†]
[‡] *www.ruizhang.info*
[†] *School of Computing and Information Systems, The University of Melbourne, Australia*
yimengd@student.unimelb.edu.au, rui.zhang@ieee.org, jianzhong.qi@unimelb.edu.au

***Abstract*—We introduce the task of webpage briefing (WB) to provide a summary of a webpage in a hierarchical manner, from the broad topic of the webpage, to finer level key attributes. A straightforward approach for this task is to train a machine learning model for generating topics and extracting key attributes. However, such a model may not perform well on webpages that are from domains not seen in the training data. An ideal model should be able to adapt to unseen domains while preserving knowledge learned from the seen domains. Knowledge distillation (KD) offers a potential solution, in which a teacher pre-trained with specific domains can pass the knowledge to a student, while unseen domains can also be added to increase the robustness of the models. However, existing works usually assume the models have no access to seen domains during distillation and the knowledge on seen domains may be lost. In our setting, we have access to the generated topics, which contain representative knowledge of seen domains and can help preserve that knowledge during distillation. Moreover, a vanilla KD does not pass on the knowledge about the location patterns of the informative contents in webpages, which are essential for identifying the topics to be generated or the key attributes to be extracted. To preserve more knowledge of seen domains and to better utilize the location patterns, we propose a Dual Distillation model which consists of identification distillation (ID) and understanding distillation (UD); ID distills knowledge on the identification of informative contents under the guidance of the learned topics of seen domains, while UD distills knowledge on topic generation or key attribute extraction. Since topics and key attributes are distilled separately in two students in Dual Distillation, the inherent correlations between them are not utilized. To better exploit such correlations, we propose a Triple Distillation model which consists of a shared ID and two UDs, one for topic generation and the other for key attribute extraction. We further propose a joint model for WB with signal enhancement and exchange among a key attribute extractor, a topic generator, and an informative section predictor. Experiments on real-world webpages show that our models achieve high performances for WB, and validate the superiority of Dual Distillation and Triple Distillation in their target settings. Experiments also show that the proposed joint model outperforms single-task baselines and other joint models.**

***Index Terms*—web mining, text analysis, knowledge distillation**

I. Introduction

The Web grows exponentially and webpage contents are becoming more complex. This results in an increasing amount of time spent on browsing webpages. Our reading speed has become a bottleneck on the amount of information that we can absorb from the Web. To increase the speed of webpage browsing, we propose a new task called *webpage briefing* (WB) and novel machine learning algorithms for the task.

TABLE I
Outputs of different tasks on the webpage in Fig. 1

Task	Task output
Webpage Briefing (Our task)	> Shopping Website for Books > Nonfiction Books > Basics of Deep Learning Project-based guide > Introduction to Deep Learning Charniak, Eugene $40.13
Webpage Summarization [1]–[4]	Introduction to Deep Learning by Eugene Charniak Hardcover Book Free Shipping!
Webpage Outline Summarization [5]	Item Overview, Similar Items, Item Description, Shipping and Payments
Text Summarization [6]–[8]	An Introduction to Deep Learning is a guide to writing deep learning programs with the widely-used Python language and TensorFlow programming environment.
Keyphrase Extraction [9], [10]	Book, Deep Learning, Python, Tensorflow
Keyphrase Generation [11]	Shipping, Product, Neural Network

WB aims to provide a summary for a webpage in a hierarchical manner such that we can quickly understand what the webpage is about and key information in it. Specifically, at the top level of the hierarchical summary is a broad topic of the webpage (e.g., a shopping website for books), followed by high-level key attributes, which may be a more precise topic or category of the webpage extracted from the webpage (e.g., nonfiction books), and then followed by more detailed and specific key attributes (e.g., Basics of Deep Learning, $40.13).

The output of WB may consist of one or two dozen of words and can be understood within a few seconds rather than minutes spent on reading a significant portion of the webpage to obtain the same information. Moreover, if a user finds the content irrelevant at a high level of the summary hierarchy, she can skip the rest. For example, if the webpage topic is sports news but the user is interested in buying sports wear, she can skip reading the rest of summary. In practice, the functionality of WB may be added to web browsers to significantly increase our webpage browsing speed. Figure 1 shows an example webpage and the corresponding WB output. Table I shows how the WB output differs from those of the existing tasks that may appear related. We will detail the differences in Section II.

A straightforward approach for this task is to train a machine learning model for generating topics and extracting

[*] Rui Zhang is the corresponding author.

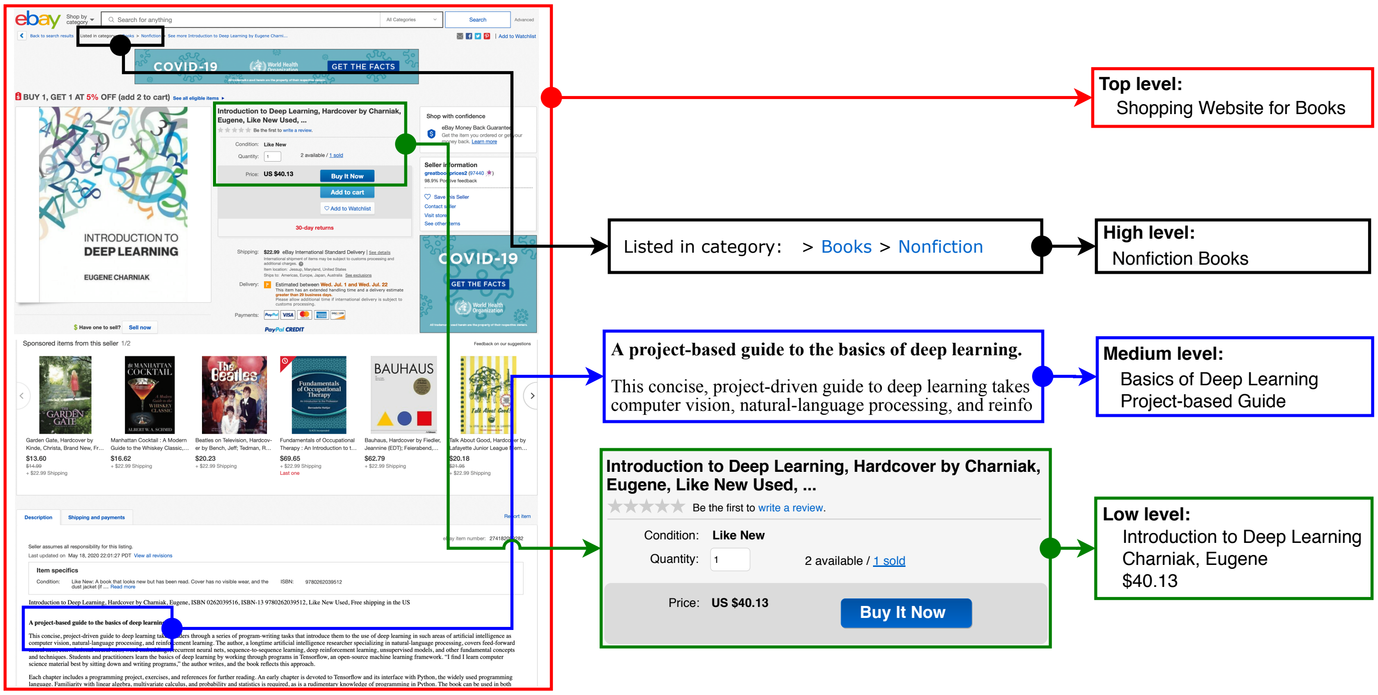

Top level:
Shopping Website for Books
Listed in category: > Books > Nonfiction
High level:
Nonfiction Books
A project-based guide to the basics of deep learning.
This concise, project-driven guide to deep learning takes computer vision, natural-language processing, and reinfo
Medium level:
Basics of Deep Learning
Project-based Guide
Introduction to Deep Learning, Hardcover by Charniak, Eugene, Like New Used, ...
Be the first to write a review.
Condition: Like New
Quantity: 1
2 available / 1 sold
Price: US $40.13
Buy It Now
Low level:
Introduction to Deep Learning
Charniak, Eugene
$40.13

Fig. 1. An example of a webpage[1]and its WB output (best viewed in color).

key attributes. However, such a model may not perform well on webpages that are from domains not seen in the training data. An ideal model for WB should be able to adapt to unseen domains while preserving knowledge learned from the seen domains. Knowledge distillation [12] (KD) offers a potential solution. Recent works [13]–[15] have demonstrated that KD can pass the knowledge in a teacher pre-trained with specific domains to a student, while unseen domains can also be added to increase the robustness of models and reduce over-fitting on seen domains. A teacher pre-trained with a variety of domains, and a randomly initialized student are trained together through KD to give prediction for unseen domains. The student gives arbitrary predictions at early distillation stage to reduce high-confidence predictions of the teacher, which are given based on the knowledge from seen domains. Then the student gradually updates during the distillation process to learn a feature space that better adapts to the context and content structure of webpages from unseen domains.

However, since existing works usually assume the student does not have access to seen domains, some knowledge may be lost during the graduate update process of distillation. They usually fine-tune the student with seen domains afterwards to recall more knowledge of seen domains or use pseudo data to help the distillation. In our setting, the topics of seen domains, which contain representative knowledge of seen domains, are stored during pre-training. Therefore, we have access to representative knowledge of seen domains, and can use them during distillation to help preserve more knowledge. Moreover, a naive adaptation of KD only guides the student to learn to generate webpage topics/attributes. It does not pass on the knowledge of the location patterns of the informative contents of webpages, which are essential for identifying the topics to be generated or the key attributes to be extracted.

To preserve more knowledge of seen domains and to utilize the location patterns of the informative contents of webpage, we propose a *Dual Distillation* (Dual-Distill) method, i.e, identification distillation and understanding distillation. A teacher is pre-trained on a large amount of labelled webpages with seen domains. A student is trained through dual distillation to generate topics *or* extract key attributes from webpages that are from unknown domains. The identification distillation guides the student to mimic the teacher's intermediate learning behavior of identifying informative sections under the guide of known topics learned from seen domains. It matches the attention distribution on a webpage between the teacher and the student towards known topics representations. Including the known topics into the distillation also brings in similarly distribution between unknown domains and seen domains, which gives additional signals to preserve the knowledge of seen domains in the intermediate distillation process. The understanding distillation guides the student to mimic the teacher's behavior of prediction through matching the output distributions between the teacher and the student.

In Dual-Distill, topics and key attributes are distilled separately in two student models, which loses the inherent correlations between the topics and key attributes in the lower level of the hierarchy. Knowing the topic of a webpage can help the prediction and extraction of key attributes. For example, in a book shopping webpage, author, title and price are more likely to be key attributes, while in a recruitment webpage, key attributes are more likely to be job, company and salary. Based on this insight, we may further improve the above method

[1] Source: https://www.ebay.com/itm/Introduction-to-Deep-Learning-by-Eugene-Charniak-Hardcover-Book-Free-Shipping-/392213703939

which requires two separate Dual-Distill for topic generation and key attributes extraction.

To better exploit such inherent correlations between topics and key attributes in the lower level of the hierarchy, we further propose a method named *Triple Distillation* (Tri-Distill), which has one *shared* identification distillation and two understanding distillations. A teacher is pre-trained to jointly extract attributes and generate topics, and a student is jointly distilled through triple distillation to generate topics *and* extract key attributes. The regularizations between two understanding distillations and the sharing in identification distillation better capture the inherent correlations.

Although the teaching (i.e., distillation) methods are important, the teacher's knowledge structure (i.e., model architecture) also influences the student's performance. We further propose a *Joint WB* model (Joint-WB) with signal enhancement and exchange mechanisms as a teacher for the proposed distillation methods. Joint-WB fully exploits the correlations between key attributes, topics and informative contents, and avoid error propagation between them. Joint-WB consists of three parts: key attribute extractor, topic generator, and informative section predictor. Informative section predictor provides signals about the location of informative sections. Key attribute extractor provides hints about informative words. Topic generator provide a fluent phrase to describe the broad topic of the webpage. To better share the learning signals among them, we add signal enhancement and exchange mechanisms during the training of three parts. In the informative section predictor, we leverage a Markov dependency mechanism to help decide the location of informative sections. In the key attribute extractor, we propose a section-and-topic dual-aware mechanism to utilize the task correlations and exchange the learning signals. In the topic generator, we use a section-and-key-attributes dual-aware mechanism to strengthen the correlations between three parst and share the signals.

Our contributions are summarized as follows:

- We propose a novel task called webpage briefing (WB), which generates a summary of a webpage in a hierarchical manner, to significantly increase the speed of browsing and comprehending webpages.
- We propose *Dual Distillation* (Dual-Distill) and *Triple Distillation* (Tri-Distill). Experimental results show the superiority of Dual-Distill and Tri-Distill in addressing their targeted problems, and outperform baselines by at most 8.63% in exact match (EM) for topic generation and 7.03% in F1 for attribute extraction on unseen domains.
- We further propose a *Joint WB* (Joint-WB) model for the whole task, which has a joint learning architecture with signal exchange and enhancement mechanisms. Experimental results show that Joint-WB achieves 95.02% in EM and 97.30% in F1 on seen domains. It outperforms single-task baselines and other jointly trained baselines.

II. Related Work

We first summarize studies on tasks related to WB. We then review two key techniques used in our proposed models, i.e., knowledge distillation and joint learning.

TABLE II
COMPARISON WITH RELATED TASKS

Task	Hierarchical	Generative	Concise	Internal content	Fluent
Webpage Briefing (Our task)	✓	✓	✓	✓	✓
Webpage Summarization [1]–[4]	✗	✗	✗	✗	✓
Webpage Outline Summarization [5]	✓	✗	✓	✓	✗
Text Summarization [6]–[8]	✗	✓	✗	✓	✓
Keyphrase Extraction [9], [10]	✗	✗	✓	✓	✗
Keyphrase Generation [11]	✗	✓	✓	✓	✗

A. Related Tasks

There are many studies towards understanding of webpages. WB has a different goal from existing study as detailed below.

Webpage summarization [1], [2] provides a textual or visual summary of a webpage. The summary comes from both internal webpage content and external knowledge about the webpage, such as text segments from other webpages that pointing to this webpage [3], or external user posts [4] about this webpage. Such external knowledge is often not available in practice. *Webpage outline summarization* [5] is a variant of webpage summarization and it uses the hierarchical HTML headings of a webpage to form a summary. Such an outline does not provide the information as provided by WB, because the headings may not reflect the contents of the webpage; they can be anything the webpage generator produces.

Text summarization [6] provides a short summary that summarizes the general meaning of a document. This task usually targets on well-formed articles, such as scientific articles [7] or news articles [8], which are not directly applicable to webpages. Moreover, it usually returns full and long sentences rather than a few words as required in WB.

Keyphrase extraction [9], [10] extracts keyphrases in webpages to capture their core topics, but it does not consider words that do not exist in the webpage. *Keyphrase generation* [11] can provide keyphrases absent from a webpage . However, both keyphrase extraction and keyphrase generation list phrases independently and isolatedly, they cannot form a piece of fluent natural language or provide the set of key attributes with a hierarchy. This may influence the understanding for essential meaning of a webpage. For example, the keyphrases *'book, online shopping'* can be interpreted as *'a book about online shopping'*, or *'a online shopping website for books'*, whose meanings are totally different.

Table II summarizes the (advantageous) characteristics of different tasks from five aspects: providing **hierarchical** summaries, providing **generative** summaries (not just extractions of words and sentences), providing **concise** summaries (not long sentences), providing summaries based on **internal content** of webpages (not relying on external knowledge), and providing summaries with **fluent** natural language. WB possesses all the advantageous characteristics whereas previous tasks have at most three of them.

B. Knowledge Distillation

Knowledge distillation (KD) [12] is originally proposed for model compression. Recent works show that KD yields a great success in preserving previously learned knowledge when learning new knowledge and confessing robustness to pre-trained models with unseen data [13]–[16]. KD has a teacher-student architecture, in which a student model is trained to mimic a pre-trained larger teacher model. In distillation, knowledge in the teacher model is transferred to the student by minimizing a loss function. The teacher model's output logits from a softmax function usually predict correct classes with a high probability, while the logits for other classes are very close to 0. This cannot provide much information beyond the ground truth when training. [17] proposes a softmax temperature to tackle this problem. It obtains soft targets from a teacher model by increasing a parameter for softmax probabilities. Our Dual-Distill and Tri-Distill follow the idea of adding a softmax temperature parameter in understanding distillation, while we further add an identification distillation to optimize intermediate learning behaviors.

C. Joint Learning

Joint learning refers to a learning paradigm that addresses multiple learning goals in a single model. It utilizes the relationship among related tasks and transfer the common knowledge across different tasks to boost the performance. This is opposed to single-task learning where individual models are learned separately, each with a single target. [18] proposes a joint learning model to learns the character features and long distance dependencies together through concatenating signals. Rather than concatenation, which may bring misleading signals, [19], [20] propose attention-based joint learning models to perform entity extraction tasks on academic homepages. Our Joint-WB is also attention-based, while it integrates the learning signals from more tasks through dual-aware attentions.

III. Proposed Models

We present Dual-Distill in Section III-A and Tri-Distill in Section III-B. We discuss Joint-WB and its signal exchange and enhancement mechanisms in Section III-C.

Table III lists the frequently used symbols. We use bold uppercase letters to denote matrices (e.g. $\mathbf{R}$) or sequences (e.g. $\mathbf{C^0}$), lowercase letters to denote scalars (e.g. r) or individual item in a sequence (e.g. c_i), and unbolded uppercase letters of different fonts to denote models (e.g. $\mathbb{T}$) or modules (e.g. $\mathcal{P}$).

Given a webpage $\mathbf{D} = \{w_1, w_2, ..., w_l\}$, where w_l is the l-th token in D, we aim to: 1) extract a set of entities $\mathbf{V} = \{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_m\}$ as the key attributes for the webpage, where $\mathbf{W}_m = \{w_i, ..., w_{i+x_m}\}$ is the m-th key attribute consisting of a sequence of x_{m+1} tokens, and 2) generate a fluent phrase $\mathbf{S} = \{s_1, s_2, ..., s_n\}$ as the topic description of the webpage, where s_n is the n-th token in the phrase.

TABLE III
Frequently Used Symbols

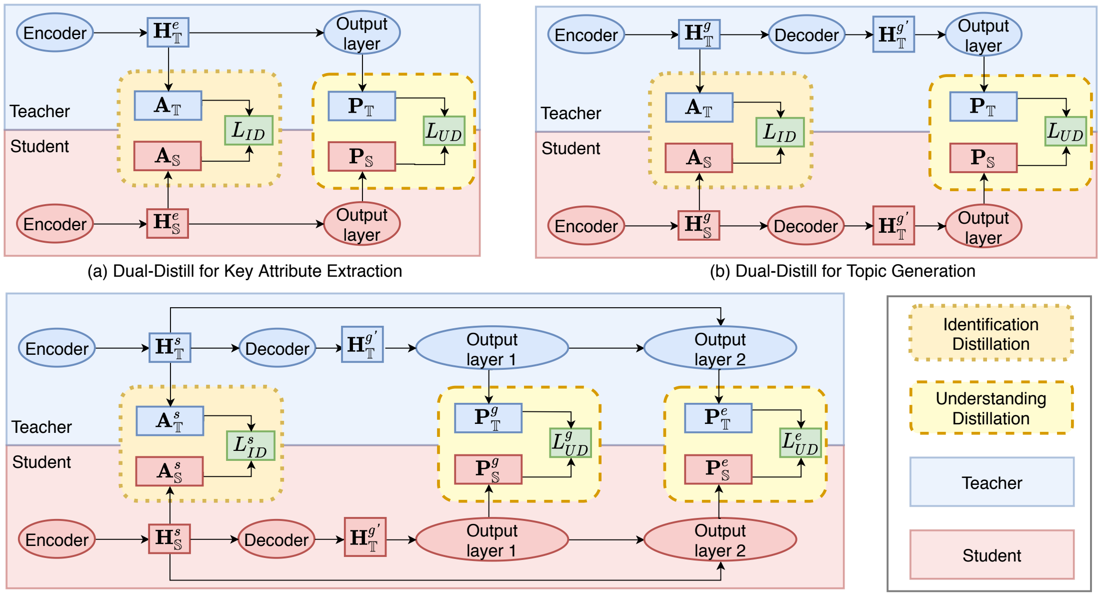
Symbol	Meaning
$\mathbb{T}$, $\mathbb{S}$	A teacher model and a student model
$\mathbf{A}_\mathbb{T}$,$\mathbf{A}_\mathbb{S}$	The attention distribution in the teacher and the student
$\mathbf{A}^s_\mathbb{T}$, $\mathbf{A}^s_\mathbb{S}$	The shared attention distribution in the teacher and the student
$\mathbf{P}_\mathbb{T}$,$\mathbf{P}_\mathbb{S}$	The output distribution in the teacher and the student
L_{ID}, L_{UD}	The identification distillation and the understanding distillation in Dual-Distill
L^s_{ID}	The shared identification distillation in Tri-Distill
L^e_{UD},L^g_{UD}	The understanding distillation of attribute extraction and topic generation in Tri-Distill
$\mathcal{E}$, $\mathcal{G}$, $\mathcal{P}$	A key attribute extractor, a topic generator and an informative section predicator
$\overleftrightarrow{\mathbf{C}}_\mathcal{E}$	The section-and-topic dual-aware hidden token representations
$\mathbf{Q}^h$	The integrated hidden topic representations
$\mathbf{C}^h_\mathcal{E}$	The section-dependent hidden token representations
$\overleftrightarrow{\mathbf{C}}_\mathcal{G}$	The section-and-key-attributes dual-aware hidden sentence representations
$\mathbf{E}^h$	The integrated hidden sentence representations
$\mathbf{C}^h_\mathcal{G}$	The section-dependent hidden sentence representations

A. Dual-Distill

An ideal WB model should be able to update its knowledge for webpages from new domains while preserving existing knowledge for seen domains, which is challenging. Knowledge distillation [17] offers a potential solution [14], [16]. However, a naive adaptation of KD does not pass on the knowledge of locating the informative contents of webpages. Moreover, the knowledge of seen domains may lost in distillation. To address the above issues, we propose a *Dual Distillation* (Dual-Distill) method which consists of identification distillation and understanding distillation, where the former distills knowledge on identifying informative contents under the guide of representative knowledge of seen domains (i.e., the topics of seen domains), and the later distills knowledge on topic generation or key attribute extraction. Fig. 2 (a) and 2 (b) illustrate the architecture of Dual-Distill.

Dual-Distill has a teacher-student architecture, in which a teacher model $\mathbb{T}$ is pre-trained on a large amount of labelled webpages $\mathbf{D}_r$ covering r topics for attribute extraction *or* topic generation, and a student model $\mathbb{S}$ is randomly initialised and trained to mimic the teacher to extract attributes *or* generate topics for new webpages $\mathbf{D}_{r+k}$, which cover $r + k$ topics and k is the number of previously unseen topics. $\mathbb{S}$ is distilled through dual distillation, i.e., identification distillation L_{ID} and understanding distillation L_{UD}.

Model details. Given pre-trained word embeddings, a new webpage D_{new} is represented as $\mathbf{C} = \{c_1, c_2, ..., c_l\}$, where c_i is the word embedding for the i-th token and l is the total number of tokens in D_{new}. The teacher model $\mathbb{T}$ encodes $\mathbf{C}$ to a hidden token representation $\mathbf{H}^e_\mathbb{T}$ for attribute extraction, or hidden sentence representation $\mathbf{H}^g_\mathbb{T}$ for topic generation using


Encoder
$\mathbf{H}^e_{\mathbb{T}}$
Output layer
Teacher
Student
$\mathbf{A}_{\mathbb{T}}$
L_{ID}
$\mathbf{A}_{\mathbb{S}}$
$\mathbf{P}_{\mathbb{T}}$
L_{UD}
$\mathbf{P}_{\mathbb{S}}$
$\mathbf{H}^e_{\mathbb{S}}$
Decoder
$\mathbf{H}^g_{\mathbb{T}}$
$\mathbf{H}^{g'}_{\mathbb{T}}$
$\mathbf{H}^g_{\mathbb{S}}$
$\mathbf{H}^s_{\mathbb{T}}$
$\mathbf{H}^s_{\mathbb{S}}$
$\mathbf{A}^s_{\mathbb{T}}$
$\mathbf{A}^s_{\mathbb{S}}$
L^s_{ID}
Output layer 1
Output layer 2
$\mathbf{P}^g_{\mathbb{T}}$
$\mathbf{P}^g_{\mathbb{S}}$
L^g_{UD}
$\mathbf{P}^e_{\mathbb{T}}$
$\mathbf{P}^e_{\mathbb{S}}$
L^e_{UD}
Identification Distillation
Understanding Distillation
Teacher
Student

(a) Dual-Distill for Key Attribute Extraction

(b) Dual-Distill for Topic Generation

(c) Tri-Distill for Key Attribute Extraction and Topic Generation

Fig. 2. The architectures of Dual-Distill (a&b) and Tri-Distill (c) (best viewed in color).

its encode layers. The student model $\mathbb{S}$ encodes $\mathbf{C}$ to hidden token representation $\mathbf{H}^e_{\mathbb{S}}$ or hidden sentence representation $\mathbf{H}^g_{\mathbb{S}}$ using its encode layers. For topic generation, $\mathbb{T}$ produces decoded hidden sentence representation $\mathbf{H}^{g'}_{\mathbb{T}}$ using the decoder layers in the teacher model; $\mathbb{S}$ produces decoded hidden sentence representation $\mathbf{H}^{g'}_{\mathbb{S}}$ using its decoder layers.

The identification distillation L_{ID} aims to guide $\mathbb{S}$ to mimic $\mathbb{T}$'s behavior of identifying informative contents from webpages under the guide of pre-defined topics representation $\mathbf{R}$ of seen domains. We achieve this by matching the attention distributions of a webpage between $\mathbb{T}$ and $\mathbb{S}$ towards the representation $\mathbf{R}$ of r pre-defined topics. The attention distributions also contain information about the similarly distribution between the webpage and seen domains, which give auxiliary signals to preserve parts of learned knowledge in $\mathbb{T}$. We minimise the sum of element-wise L1 difference between the normalized attention distributions $\mathbf{A}_{\mathbb{T}}$ of $\mathbb{T}$ and the normalized attention distribution $\mathbf{A}_{\mathbb{S}}$ of $\mathbb{S}$ over r pre-defined topics:

$$L_{ID} = \sum_{i=1}^{r} \| \frac{\mathbf{A}^i_{\mathbb{T}}}{\|\mathbf{A}^i_{\mathbb{T}}\|_2} - \frac{\mathbf{A}^i_{\mathbb{S}}}{\|\mathbf{A}^i_{\mathbb{S}}\|_2} \|_1$$

For the key attribute extraction task, the attention distribution $\mathbf{A}_{\mathbb{T}}$ records the relationship between $\mathbf{H}^e_{\mathbb{T}}$ and the topic phrase matrix $\mathbf{R}$ of the r pre-defined topics, and the attention distribution $\mathbf{A}_{\mathbb{S}}$ records the relationship between $\mathbf{H}^e_{\mathbb{S}}$ and the topic phrase matrix $\mathbf{R}$ of r pre-defined topics:

$$\mathbf{A}_{\mathbb{T}} = softmax(\mathbf{H}^e_{\mathbb{T}} \mathbf{W}_{AT} \mathbf{R}^{\top})$$
$$\mathbf{A}_{\mathbb{S}} = softmax(\mathbf{H}^e_{\mathbb{S}} \mathbf{W}_{AS} \mathbf{R}^{\top})$$

For the general topic generation task, the attention distribution $\mathbf{A}_{\mathbb{T}}$ records the relationship between $\mathbf{H}^g_{\mathbb{T}}$ and the topic phrase matrix $\mathbf{R}$ of r pre-defined topics, and the attention distribution $\mathbf{A}_{\mathbb{S}}$ records the relationship between $\mathbf{H}^g_{\mathbb{S}}$ and the topic phrase matrix $\mathbf{R}$ of r pre-defined topics:

$$\mathbf{A}_{\mathbb{T}} = softmax(\mathbf{H}^g_{\mathbb{T}} \mathbf{W}_{AT} \mathbf{R}^{\top})$$
$$\mathbf{A}_{\mathbb{S}} = softmax(\mathbf{H}^g_{\mathbb{S}} \mathbf{W}_{AS} \mathbf{R}^{\top})$$

where $\mathbf{W}_{AT}$ and $\mathbf{W}_{AS}$ are trainable parameters. The topic phrase matrix $\mathbf{R}$ is the concatenation of r previously seen topic phrases. Each topic phrase is represented by combining all the words in a topic phrase. We concatenate the hidden representations of all the tokens in a topic phrase, which are learned in the pre-trained tearcher model $\mathbb{T}$. We pass the concatenated representations through a dense layer with $tanh$ as a non-linear activation function:

$$\mathbf{R} = R_1 \oplus R_2 \oplus ... \oplus R_r$$
$$R_i = tanh((q^1_i \oplus q^2_i \oplus ... \oplus q^{n_i}_i) W_{R_i})$$

where R_i is the representation of the i-th topic phrase representation, $i \in [1, r]$, W_{R_i} is a trainable parameter, $q^{n_i}_i$ is the n_i-th hidden token representation in the i-the topic phrase, and n_i is the length of the i-th topic phrase.

The understanding distillation L_{UD} aims to guide $\mathbb{S}$ to mimic $\mathbb{T}$'s behavior of doing prediction. We achieve this by matching the output distributions between $\mathbb{T}$ and $\mathbb{S}$. Specifically, we minimise the Kullback-Leibler divergence between the output distribution $\mathbf{P}_{\mathbb{T}}$ of $\mathbb{T}$ and the output distribution $\mathbf{P}_{\mathbb{S}}$ of $\mathbb{S}$:

$$L_{UD} = \sum \mathbf{P}_{\mathbb{T}} \log(\frac{\mathbf{P}_{\mathbb{T}}}{\mathbf{P}_{\mathbb{S}}})$$

For the key attribute extraction task, the output distributions $\mathbf{P}_{\mathbb{T}}$ and $\mathbf{P}_{\mathbb{S}}$ are computed using the hidden token representation

$\mathbf{H}^e_{\mathbb{T}}$ and $\mathbf{H}^e_{\mathbb{S}}$, respectively, with a softmax temperature γ [17]:

$$\mathbf{P}_{\mathbb{T}} = softmax(\frac{\mathbf{H}^e_{\mathbb{T}}\mathbf{W}_{PT} + b_T}{\gamma})$$
$$\mathbf{P}_{\mathbb{S}} = softmax(\frac{\mathbf{H}^e_{\mathbb{S}}\mathbf{W}_{PS} + b_S}{\gamma})$$

For the general topic generation task, the output distributions $\mathbf{P}_{\mathbb{T}}$ and $\mathbf{P}_{\mathbb{S}}$ are computed using the decoded hidden token representation $\mathbf{H}^{g'}_{\mathbb{T}}$ and $\mathbf{H}^{g'}_{\mathbb{S}}$, respectively, with a softmax temperature γ [17]:

$$\mathbf{P}_{\mathbb{T}} = softmax(\frac{\mathbf{H}^{g'}_{\mathbb{T}}\mathbf{W}_{PT} + b_T}{\gamma})$$
$$\mathbf{P}_{\mathbb{S}} = softmax(\frac{\mathbf{H}^{g'}_{\mathbb{S}}\mathbf{W}_{PS} + b_S}{\gamma})$$

where $\mathbf{W}_{PT}$ and $\mathbf{W}_{PS}$ are trainable parameters.

Then, Dual-Distill is trained by minimising the sum of total loss L with parameters α and γ:

$$L = \alpha L_{ID} + \gamma^2 L_{UD}$$

where γ^2 is set for L_{UD} following [17] since the magnitudes of gradients produced by L_{UD} scale as $1/\gamma^2$.

Although we illustrate Dual-Distill with two levels output for the WB hierarchy, Dual-Distill can be seen as a general framework and can be extended to provide more than two levels of output. Towards more levels of output for the WB hierarchy, a teacher model $\mathbb{T}$ at each level can be pre-trained and a student model $\mathbb{S}$ can be distilled using the Dual-Distill framework. The time complexity of training Dual-Distill depends on the choice of teacher and student models. We use t_t, t_s to denote the time cost of a single teacher model and a single student model, b to denote the batch size, n to denote the sequence length in each batch, r to denote the number of previously seen topic phrases and g to denote the length of the generated topic phrase. The time complexity of training Dual-Distill is $O(b\times(t_t+t_s+nr+n))$ for key attribute extraction and $O(b \times (t_t + t_s + nr + g))$ for topic generation.

B. Tri-Distill

In Dual-Distill, the general topic and key attributes of a webpage are distilled separately in two student models, which loses the inherent correlations between key attributes and topics. To better exploit such inherent correlations, we further propose a method named *Triple Distillation* (Tri-Distill), which has one *shared* identification distillation and two understanding distillations, one for attribute extraction *and* the other for topic generation. Fig. 2 (c) shows the architecture of Tri-Distill.

Tri-Distill has a similar teacher-student architecture to that of Dual-Distill. The difference is that, in Tri-Distill, a teacher model $\mathbb{T}$ is pre-trained to jointly extract attributes *and* generate topics. A student model $\mathbb{S}$ is jointly distilled across the attribute extraction task *and* the topic generation task through one shared identification distillation L^s_{ID}, and two understanding distillation L^e_{UD} and L^g_{UD}. The regularizations between L^e_{UD} and L^g_{UD} and the sharing in L^s_{ID} lead to more universal hidden representations for two tasks, and better utilize the inherent correlations between two tasks during distillation.

Model details. The two understanding distillations L^e_{UD} and L^g_{UD} are computed using the same methods as described in Dual-Distill for the key attribute extraction task and the general topic generation task, respectively.

The shared identification distillation L^s_{ID} is computed based on the shared attention distributions $\mathbf{A}^s_{\mathbb{S}}$ and $\mathbf{A}^s_{\mathbb{T}}$:

$$L^s_{ID} = \sum_{i=1}^{N} \| \frac{\mathbf{A}^{s\,i}_{\mathbb{T}}}{\|\mathbf{A}^{s\,i}_{\mathbb{T}}\|_2} - \frac{\mathbf{A}^{s\,i}_{\mathbb{S}}}{\|\mathbf{A}^{s\,i}_{\mathbb{S}}\|_2} \|_1$$

where the shared attention distributions $\mathbf{A}^s_{\mathbb{S}}$ and $\mathbf{A}^s_{\mathbb{T}}$ are computed using the same methods as described in Dual-Distill with shared hidden token representations $\mathbf{H}^e_{\mathbb{S}}$ and $\mathbf{H}^e_{\mathbb{T}}$ obtained from a shared encoder layer.

Then, Tri-Distill is trained by minimising the sum of total loss L with parameters λ, μ, ν, and γ:

$$L = \lambda L^s_{ID} + \mu\gamma^2 L^e_{UD} + \nu\gamma^2 L^g_{UD}$$

Tri-Distill can also be extended to provide more than two levels of output for the WB hierarchy. Towards more levels of output for the WB hierarchy, a joint teacher model $\mathbb{T}$, which jointly generates topics and extracts multiple levels of attributes, can be pre-trained, and a student model $\mathbb{S}$ can be distilled using the Tri-Distill method to jointly distilled across the topic generation, and different levels of attribute extraction. We use t_t, t_s to denote the time cost of a single teacher model and a single student model, b to denote the batch size, n to denote the sequence length in each batch, r to denote the number of previously seen topic phrases and g to denote the length of the generated topic phrase. Then the time complexity of training Tri-Distill is $O(b \times (t_t + t_s + nr + n + g))$.

C. Joint-WB

So far we have focused on the teaching (i.e., distillation) process in KD. Although the teaching methods are important, the teacher model's knowledge structure (i.e., model architecture) also influences the student model's performance. We propose Joint-WB as a powerful teacher for the proposed distillation methods. Joint-WB fully exploits the correlations between key attributes, topics and informative contents, and avoid error propagation between them. It consists of three parts: a key attribute extractor $\mathcal{E}$, a topic generator $\mathcal{G}$, and an informative section predicator $\mathcal{P}$. Fig. 3 illustrates the architecture of Joint-WB.

To better capture and share the learning signals between $\mathcal{E}$, $\mathcal{G}$ and $\mathcal{P}$, we further add signal enhancement and exchange mechanisms during the training of three parts. Specifically, $\mathcal{P}$ predicts the location boundaries of informative sections through a Markov dependency mechanism. $\mathcal{E}$ extracts key attributes from a webpage using a section-and-topic dual-aware signal exchange mechanism, and $\mathcal{G}$ generates a piece of fluent text in natural language to summarize the general topic of the webpage using a section-and-key-attributes dual-aware signal exchange mechanism.

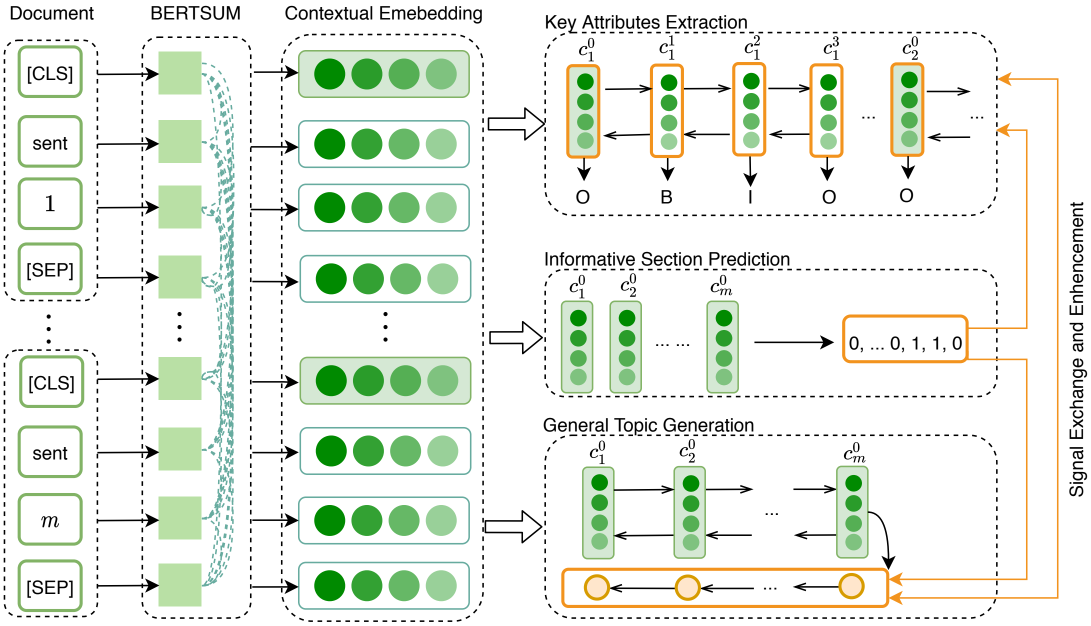


Fig. 3. The architectures of Joint-WB

Model details. Given a webpage document $D = \{w_1^0, w_1^1, ..., w_1^{n_1}, ..., w_m^0, w_m^1, ..., w_m^{n_m}\}$, where w_j^i is the i-th token in the j-th sentence, m is the number of sentences, n_j is the length of the j-th sentence, and w_j^0 is the j-th [CLS] symbol inserted to indicate the start of the j-th sentence, which helps collect latent summarizing features of the sentence [21]. Joint-WB uses BERTSUM [21] to encode D to a sequence of vectors $\mathbf{C} = \{c_1^0, c_1^1, ..., c_1^{n_1}, ..., c_m^0, c_m^1, ..., c_m^{n_m}\}$, where c_j^i is the contextual embedding of the i-th token in the j-th sentence, and c_j^0 is the contextual embedding of the j-th sentence. The contextual embedding of all the sentences in D is denoted as $\mathbf{C}^0 = \{c_1^0, c_2^0, ..., c_m^0\}$.

The key attribute extractor $\mathcal{E}$ converts $\mathbf{C}$ to hidden token representations $\mathbf{C}_\mathcal{E} = \{e_1, e_2, ..., e_l\}$ using Bi-LSTM [22], where l is the total number of tokens and e_i is the i-th hidden token representation. The topic generator $\mathcal{G}$ employs an encoder-decoder framework [23]. It converts $\mathbf{C}^0$ to hidden sentence representations $\mathbf{C}_\mathcal{G} = \{g_1, g_2, ..., g_m\}$ using Bi-LSTM [22] and use LSTM [22] to decode and generate hidden topic representations $\mathbf{Q} = \{q_1, q_2, ..., q_{n'}\}$, where $n^{'}$ is the length of the topic phrase, q_i is the i-th hidden topic token representation in the generated topic phrase. The informative section predictor $\mathcal{P}$ converts $\mathbf{C}^0$ to informative section vectors $\mathbf{C}_\mathcal{P}^0 = \{p_1, ..., p_m\}$ by a Markov dependency mechanism, i.e., to decide whether the j-th sentence is in an informative section by looking at the $j-1$ and $j+1$ sentences:

$$p_j = \begin{cases} 1, & sigmoid(c_{j-1}^0 W_p^1 c_j^0 + c_j^0 W_p^2 c_{j+1}^0) \geqslant 0.5 \\ 0, & sigmoid(c_{j-1}^0 W_p^1 c_j^0 + c_j^0 W_p^2 c_{j+1}^0) < 0.5 \end{cases}$$

where W_p^1 and W_p^2 are trainable parameters. $p_j = 1$ means the sentence is in an informative section and $p_j = 0$ otherwise.

To fully exploit the correlation of attributes, topics and informative sections, we share the learning signals of $\mathcal{E}$, $\mathcal{G}$ and $\mathcal{P}$ through signal exchange and enhancement mechanisms. The hidden token representations $\mathbf{C}_\mathcal{E}$ is updated to yield section-and-topic dual-aware hidden token representations $\overleftrightarrow{\mathbf{C}_\mathcal{E}}$ using an attention $\mathbf{A}_\mathcal{E}$ between the section and the topic:

$$\overleftrightarrow{\mathbf{C}_\mathcal{E}} = \mathbf{C}_\mathcal{E} \mathbf{A}_\mathcal{E}$$

Attention $\mathbf{A}_\mathcal{E}$ records the relationship between the integrated hidden topic representations $\mathbf{Q}^h$ and the section-dependent hidden token representations $\mathbf{C}_\mathcal{E}^h$:

$$\mathbf{A}_\mathcal{E} = softmax(\mathbf{C}_\mathcal{E}^h \mathbf{W}_{AE} {\mathbf{Q}^h}^\top)$$

where $\mathbf{W}_{AE}$ is trainable parameter. $\mathbf{Q}^h$ is the combination of all the information in the generated topic phrase. We concatenate the hidden representations of all the tokens in $\mathbf{Q}$ and pass it through a dense layer with $tanh$ as a non-linear activation function. $\mathbf{C}_\mathcal{E}^h$ is the update of $\mathbf{C}_\mathcal{E}$, which contains section information. We concatenate $\mathbf{C}_\mathcal{E}$ with an injected section distribution, and pass it through a dense layer with $tanh$ as a non-linear activation function:

$$\mathbf{Q}^h = tanh((q_1 \oplus q_2 \oplus ... \oplus q_{n'})\mathbf{W}_Q)$$
$$\mathbf{C}_\mathcal{E}^h = tanh((\mathbf{C}_\mathcal{E} \oplus \Phi_\mathcal{E}(p_j))\mathbf{W}_{CE})$$

where $\mathbf{W}_Q$ and $\mathbf{W}_{CE}$ are trainable parameters. $\Phi_\mathcal{E}$ is a function which injects p_j into the same dimensions as $\mathbf{C}_\mathbf{E}$.

The hidden sentence representation $\mathbf{C}_\mathcal{G}$ is updated to yield a section-and-key-attributes dual-aware hidden representation $\overleftrightarrow{\mathbf{C}_\mathcal{G}}$ using an attention $\mathbf{A}_\mathcal{G}$ between the section and key attributes through:

$$\overleftrightarrow{\mathbf{C}_\mathcal{G}} = \mathbf{C}_\mathcal{G} \mathbf{A}_\mathcal{G}$$

Attention $\mathbf{A}_\mathcal{G}$ records the relationship between the integrated hidden token representations $\mathbf{E}^h$ and the section-dependent hidden sentence representations $\mathbf{C}_\mathcal{G}^h$:

$$\mathbf{A}_\mathcal{G} = softmax((\mathbf{C}_\mathcal{G}^h \odot \mathbf{E}^h)\mathbf{W}_{AG})$$

where $\mathbf{W}_{AG}$ is trainable parameter and $\odot$ is element-wise multiplication. $\mathbf{E}^h$ contains the informations of key attributes. We concatenate the hidden token representations in $\mathbf{C}_{\mathcal{E}}$ and pass it through a dense layer with $tanh$ as a non-linear activation function. $\mathbf{C}^h_{\mathcal{G}}$ is the update of $\mathbf{C}_{\mathcal{G}}$, which contains section information. We concatenate $\mathbf{C}_{\mathcal{G}}$ with an injected section distribution, and pass it through a dense layer with $tanh$ as a non-linear activation function:

$$\mathbf{E}^h = tanh((e_1 \oplus e_2 \oplus ... \oplus e_l)\mathbf{W}_E)$$
$$\mathbf{C}^h_{\mathcal{G}} = tanh((\mathbf{C}_{\mathcal{G}} \oplus \Phi_{\mathcal{G}}(p_j))\mathbf{W}_{CG})$$

where $\mathbf{W}_E$ and $\mathbf{W}_{CG}$ are trainable parameters. $\Phi_{\mathcal{G}}$ is a function which injects p_j into the same dimensions as $\mathbf{C}_{\mathbf{G}}$.

Then, $\overleftrightarrow{\mathbf{C}_{\mathcal{E}}}$ is fed into a softmax based output layer to get the output distribution $\mathbf{O}_e$, $\overleftrightarrow{\mathbf{C}_{\mathcal{G}}}$ is firstly fed into the decoder layer then a softmax based output layer to produce the output distribution $\mathbf{O}_g$. Joint-WB is trained to jointly minimise the total loss L:

$$L = CrossEntropy(\mathbf{O}_e, gt_e) + CrossEntropy(\mathbf{O}_g, gt_g)$$

where gt_e and gt_g are ground truth and $CrossEntropy$ is a function that computes cross-entropy loss.

Joint-WB is built on the BERT$_{\text{base}}$ model. The time complexity of training Joint-WB is impacted by BERT$_{\text{base}}$. We use l to denote the maximum document length, b to denote the maximum batch size, t_b to denote the time cost of BERT$_{\text{base}}$ for each batch, i to denote the number of sentences in a sequence, d to denote the dimensionality of hidden states. The time complexity of training Tri-Distill is $O(\frac{bl}{4} \times (t_b + d^2 + ld + d + 1))$. To extend the Joint-WB model to more than two levels of hierarchy, we can use multiple extractors $\mathcal{E}$ to tackle key attributes at different levels, combine the signals from different levels, and share the combined signals with the generator $\mathcal{G}$. We leave an in-depth study for future work.

IV. Experiments

We first describe the experimental setup in Section IV-A. We evaluate Dual-Distill and Tri-Distill in Section IV-B and Joint-WB in Section IV-C. In Section IV-E, we inspect the model sensitivity on synthetic webpages and perform a human evaluation over the model outputs. Even though the WB results may be a hierarchy of multiple levels, we run experiments with two levels because the labelled data is two levels. We leave experimental study on more levels to future work.

A. Experimental Setup

1) Dataset Construction: We constructed a dataset consisting of around 655K English webpages collected from 312 websites. Among the 655K webpages, 620k webpages (D$_{\text{jasm}}$) are collected from 305 websites based on the Jasmine Directory[2], which is a web directory organised in topic based categories. The collected webpages cover 153 topics, and each topic contains two websites. For each website, 1,500 to 2,000 content-rich webpages are downloaded using the structure-driven crawler [24]. Indexing webpages and multimedia webpages such as video, music and image pages are not included. Another 30k webpages (D$_{\text{swde}}$) are collected based on the webpage list provided by the SWDE dataset [25], which contains labelled key attributes for different content-rich webpages. These 30k webpages cover seven websites and seven topics, and 1,500 to 2,200 webpages are downloaded for each website. Overall, the averaged webpage length is 1731.6 (std=210.3) tokens, and the total vocabulary size is 13M. The number of attributes in each webpage is four, and the averaged topic length is three (std=0.74) tokens.

2) Dataset Quality: We randomly select 500 webpages and ask five volunteers to assign a score of 2 (perfectly suitable/correct), 1 (suitable/correct), or 0 (unsuitable/incorrect) to each webpage to indicate: *i)* whether the webpage is content-rich, *ii)* whether the given topic suitably summarizes the general idea of the webpage, and *iii)* whether the given attributes are correct. All the volunteers have studied English for at least ten years and are trained on the scoring criteria for 25 minutes. We compute the inter-annotator agreement using Cohen's κ measurement. The result shows that the volunteers have very high agreement (κ >0.93) for all three evaluation aspects. All the webpages are content-rich based on a majority vote. All the topics suitably summarize the webpages, among which 92.6% are perfectly suitable. All the webpages have correctly labelled attributes. This demonstrates that the trustworthiness of our constructed dataset.

3) Preprocessing: For all the webpages, we use an open-source automated rendering software[3] to render the webpages and collect visible texts from the webpages. We convert all the text into lowercase and replace digits with token <*digit*>. The text is tokenised with BERT's WordPieces tokenizer where each newline character, <*digit*>, and punctuation is preserved as a single token. We follow the document representation method [21] to insert [CLS] tokens at the start of each sentence. Each document is zero-padded to the same length of 2,048 and is splitted into four 512 sub-documents because of the input length limitation of BERT.

4) Evaluation Metrics: We use precision (P), recall (R), and F1-score (F1) to evaluate the performance of key attribute extraction. For topic generation, we report both exact matching (EM) and relaxed matching (RM) performance. In exact matching, a generated topic is considered correct only if it exactly matches the ground truth. In relaxed matching, a generated topic is considered correct if it contains at least one token of the ground truth. McNemar's test of p<0.05 is used to test whether the improvements are statistically significant.

5) Implementation Details: Our proposed Dual-Distill and Tri-Distill methods with Joint-WB as the teacher are built on the BERT$_{\text{base}}$ model and are trained on GTX 1080 GPUs. All our methods are optimized using an Adam with $\beta 1$ = 0.9 and $\beta 2$ = 0.999, an initial learning rate of 0.1 with decay rate of 0.1, gradient clipping = 0.1, and we use a linear warm-up strategy with 2,000 warm-up steps. The batch size

[2] https://www.jasminedirectory.com/

[3] https://www.seleniumhq.org/

is set to 16 with a sub-document length of 512 for BERT because of the input length limitation. The batch size is set to 4 with a document length of 2,048 to preserve the intra-document relation for general topic generation and key attribute extraction. All the LSTM hidden states are set to 108 dimensions, and a dropout rate of 0.2 is applied to avoid overfitting. We use beam search in the inference process, the size and the depth of which are set to 200 and 4, respectively. The hyperparameters α is set to 0.1, γ is set to 2, λ is set to 0.1, μ is set to 1 and ν is set to 2.25. All the hyperparameters are tuned using the same development data. The training is early stopped once convergence is determined on the development dataset. Joint-WB takes about 23 hours (9 epochs) to converge. Dual-Distill takes about 3.5 hours to converge (3 epochs). Tri-Distill takes about 3.8 hours to converge (3 epochs).

6) Baselines and variants for Joint-WB: i) Single-task baselines. To evaluate whether joint training of sub-tasks can help the learning process, we compare Joint-WB with single-task models:

- ***↦Bi-LSTM**: we use Bi-LSTM [22] as encoder for key attribute extraction;
- ***↦[Bi-LSTM, LSTM]**: we use Bi-LSTM as encoder and LSTM as decoder for topic generation.

Here, *↦ represents using different word embedding methods in single-task models, e.g., **GloVe↦[Bi-LSTM, LSTM]**.

We compare both context independent and context dependent word embedding methods in single-task models to evaluate how much improvements are caused by word embeddings:

- **GloVe↦***: we directly apply context independent word embeddings learned by GloVe [26];
- **BERT↦***: we fine-tune BERT [27] to learn context dependent word embeddings;
- **BERTSUM↦***: we fine-tune BERTSUM [21] to learn context dependent word embeddings.

Here, ↦* represents applying word embedding methods to different single-task models, e.g., **BERT↦Bi-LSTM**.

We also add different prior knowledge to single-task models to evaluate whether adding signals about informative sections and topics can help the learning process:

- **+prior section**: we add prior knowledge about informative sections to Bi-LSTM for both tasks following the concatenation procedure in ATAE-LSTM [28];
- **+prior topic**: we add prior knowledge about topics to Bi-LSTM for key attribute extraction following the concatenation procedure in ATAE-LSTM [28].

ii) Joint variants and baselines. To evaluate the effectiveness of our joint model and the signal exchange and enhancement mechanisms, we compare Joint-WB with a **no signal exchange** method, **concat-based signal exchange** methods, **attention-based signal exchange** methods, and a **section signals enhancement** method:

- **Naive-Join**: we directly train two baseline extractor and generator together by minimising the total loss;
- **Con-Extractor**: we concate the hidden token representation in extractor with the topic representations from a baseline generator following [18], and jointly train the extractor and generator by minimising the total loss;
- **Ave-Extractor**: we concate the hidden token representation in extractor with the average of topic representations from a baseline generator following [18], and jointly train the extractor and generator by minimising the total loss;
- **Att-Extractor**: we implement a topic-aware extractor following the implementation of dual-aware hidden token representation learning in Section III-C without the section-aware part when computing attention, and jointly train the topic-aware extractor with a basic generator by minimising the total loss;
- **Att-Extractor+Att-Generator**: we implement a key-attributes-aware generator following the implementation of dual-aware hidden representation learning in Section III-C without the section-aware part when computing attention, and jointly train topic-aware extractor and key-attributes-aware generator by minimising the total loss;
- **Pip-Extractor+Pip-Generator**: we implement a Pip-Extractor, which has a pipeline architecture of topic-dependent and section-dependent representations learning. The topic-dependent representation is get from removing the section-aware part when computing dual-aware attention and the section-dependent representation is get from removing the topic-aware part when computing dual-aware attention. Similarly, we implement a Pip-Generator. We jointly train Pip-Extractor andPip-Generator by minimising the total loss.

7) Variants for Dual-Distill and Tri-Distill: i) Distillation model variants. To evaluate the effectiveness of dual distillation, we compare Dual-Distill with different distillation methods. We use Joint-WB as the teacher model in different variants.

- **No Distill**: we directly apply a pre-trained model on new webpages from unknown domains;
- **ID only**: we remove the understanding distillation in Dual-Distill and only apply identification distillation.
- **UD only**: we remove the identification distillation in Dual-Distill and only apply understanding distillation.
- **Pip-Distill:** we implement a pipeline method with two Dual-Distill, in which we feed the output of the first Dual-Distilled student for topic generation as a prior knowledge when Dual-Distilling the second student for attribute extraction. The generated topic from the first student is fed to the second following the topic-aware representation learning in Att-Extractor and Pip-Extractor.

ii) Teacher models variants. To evaluate the applicability and effectiveness of Dual-Distill and Tri-Distill with different teacher models, we apply them on both single-task and joint teacher models. Notice that Dual-Distill only distills one task when applied on joint teacher models, while Tri-Distill jointly distills across two tasks when applied on joint teacher models.

- **BERT-Single**: we apply Dual-Distill on BERTSUM↦Bi-LSTM for attribute extraction, and on BERTSUM↦[Bi-LSTM, LSTM] (cf. Section IV-A6-i) for topic generation;

TABLE IV
RESULTS OF DIFFERENT DISTILLATION METHODS FOR TOPIC GENERATION.

Methods	Unseen domains		Seen domains		All	
	EM	RM	EM	RM	EM	RM
No Distill	86.23	89.10	95.02	97.24	90.17	93.23
ID only	94.26	95.82	95.03	97.24	94.73	97.02
UD only	94.40	95.98	94.85	97.19	94.68	96.98
Dual-Distill	**94.86**	**96.10**	94.98	97.24	**94.93**	**97.13**

TABLE V
RESULTS OF USING DIFFERENT TEACHER MODELS IN DUAL-DISTILL AND TRI-DISTILL ON PREVIOUSLY UNSEEN DOMAINS.

Methods	BERT-Single		Naive-Join		Joint-WB	
	F1	EM	F1	EM	F1	EM
No Distill	44.10	77.23	47.23	81.84	51.21	86.23
Dual-Distill	50.79	**85.18**	53.10	**89.27**	57.28	**94.86**
Pip-Distill	51.55	-	54.02	-	58.02	-
Tri-Distill	-	-	**54.26**	∗	**58.20**	∗

Note: We do not expect the results marked with ∗ to be better because these are not intended for the target sub-task.

- **Naive-Join**: we apply Dual-Distill and Tri-Distill on Naive-Join model (cf. Section IV-A6-ii);
- **Joint-WB**: we apply Dual-Distill and Tri-Distill on Joint-WB model (cf. Section III-C).

B. Evaluation of Dual-Distill and Tri-Distill

To evaluate Dual-Distill and Tri-Distill, we use webpages from 140 topics to train the teacher models, and use webpages from another 20 topics to train the Dual-Distill and Tri-Distill. The dataset is randomly taken from $\mathrm{D_{swde}}$ and $\mathrm{D_{jasm}}$ following 80%-10%-10% train-develop-test splits, respectively.

Tables IV and V summarize the results. Overall, Dual-Distill outperforms baseline distillation methods on webpage from unknown domains, which may contain both previously seen domains and unseen domains. Applying Dual-Distill and Tri-Distill with different teacher models on webpages from previously unseen domains outperforms direct applying trained teacher models on these webpages. Tri-Distill outperforms the pipeline of two Dual-Distill for attribute extraction.

1) Effectiveness of Distillation: Table IV reports the results where we compare our Dual-Distill method with different distillation methods for topic generation. Overall, the distilled models outperform direct applying the pre-trained teacher model (No Distill) on webpages with previous unseen topics by at most 8.63% in EM and 7.00% in RM. All the distilled models achieve similar performance as direct applying the pre-trained model on webpages with previous seen topics. These results indicate that the distilled models retain the previous learned information from the pre-trained teacher model, whilst updating the student model to learn from the new webpages.

Both identification distillation (ID) and understanding distillation (UD) are critical to the performance of dual distillation on previously unseen domains, i.e., removing either would result in a drop in performance. UD is more important than ID for previously unseen domains, i.e., performance drops more when UD is removed. Dual-Distill outperforms baseline methods on webpage from unknown domains, which may contain both previously seen domains and unseen domains by at most 4.76% in EM and 3.90% in RM.

TABLE VI
RESULTS OF SINGLE-TASK MODELS FOR ATTRIBUTE EXTRACTION USING DIFFERENT WORD EMBEDDING AND PRIOR KNOWLEDGE.

Methods	P	R	F1
No prior topic			
GloVe↦Bi-LSTM	96.28	83.73	89.57
BERT↦Bi-LSTM	95.00	90.12	92.50
BERTSUM↦Bi-LSTM	95.02	90.12	92.51
BERTSUM↦Bi-LSTM + prior section	95.11	91.47	93.25
+ prior topic			
GloVe↦Bi-LSTM	98.47	89.12	93.56
BERT↦Bi-LSTM	98.14	94.21	96.14
BERTSUM↦Bi-LSTM	98.17	94.23	96.16
BERTSUM↦Bi-LSTM + prior section	98.14	95.62	96.86
Joint-WB (our proposed)	**98.42**	**96.21**	**97.30**

2) Applicability of Distillation: Table V reports the results where we compare the performance on previously unseen domains using different distillation methods with different teacher models. Overall, using Dual-Distill with different teacher models can improve the performance of both tasks. This may be explained by the dual distillation, which enhances the student model's ability of discriminating between webpage topics to locate similar attributes. Feeding the output of the topic generation as a prior knowledge to the attribute extraction in a pipeline manner (Pip-Distill) improves the performance of attribute extraction in F1 score. This is because distillation with topic information brings stronger hints about the contents related to a topic and reduce the wrong extraction of attributes from irrelevant contents. Tri-Distill further outperforms Pip-Distill for attribute extraction. This could be explained by the shared identification distillation, which leads to a more general attention distribution and better hidden token representations.

The overall performance of attribute extraction is not as good as the overall topic generation performance. This is expected as the key attributes are difficult to extract without any pre-defined attribute types for webpages on different topics. Tri-Distill achieves 77.49% in EM when using the naive-join model, and 82.82% in EM when using the Joint-WB model. The performance of Tri-Distill for topic generation is not as good as direct applying the teacher model (No Distill) or Dual-Distill to topic generation only, which is the same as we expect. This could be explained by the misleading signals from key attributes during the joint distillation, which drag down the performance of topic generation.

C. Evaluation of Joint-WB

We compare the performance of Joint-WB with baseline models on webpages with previously seen domains. We use data from $\mathrm{D_{swde}}$ and $\mathrm{D_{jasm}}$ following the random 80%-10%-10% train-develop-test splits. Tables VI, VII, VIII and IX present the results. Overall, Joint-WB outperforms single-task baselines and all joint baselines for both tasks.

TABLE VII
RESULTS OF SINGLE-TASK MODELS FOR TOPIC GENERATION USING DIFFERENT WORD EMBEDDING AND PRIOR KNOWLEDGE.

Methods	EM	RM
Glove↦[Bi-LSTM, LSTM]	85.38	86.86
BERT↦[Bi-LSTM, LSTM]	91.54	93.04
BERTSUM↦[Bi-LSTM, LSTM]	91.63	93.28
BERTSUM↦[Bi-LSTM, LSTM] + prior section	92.20	93.81
Joint-WB (our proposed)	**95.02**	**97.24**

1) Comparison with Single-task Baselines: Tables VI and VII report the performance of single-task models. Overall, Joint-WB outperforms the baselines on both attribute extraction and topic generation by considerable margins. It outperforms the best baseline without any prior knowledge by 4.79% in F1 for attribute extraction (Table VI) and by 3.40% in EM for topic generation (Table VII). This can be explained by the joint learning and signal exchange mechanisms used in our model, which help capture the inherent correlations between the subtasks, and lead to better hidden representations as well as better prediction results.

Among the baseline models, as Table VI shows, for attribution extraction, knowing the topics of webpages helps improve the performance with at least 3.64% in F1 (No prior topic BERT↦Bi-LSTM vs. +prior topic BERT↦Bi-LSTM). This confirms the intuition that knowing the general topic of a webpage can guide locating the key attributes.

The baseline models yield better results when the informative sections of webpages are given as prior knowledge, i.e., BERTSUM↦Bi-LSTM + prior section outperforms BERTSUM↦Bi-LSTM by 0.74% in F1 for attribute extraction, and BERTSUM↦[Bi-LSTM, LSTM] + prior section outperforms BERTSUM↦[Bi-LSTM, LSTM] by 0.57% in EM for topic generation. This confirms the intuition that knowing the informative section can provide signals about the general location of important contents and reduce the influence of less informative contents.

For both tasks, the BERT based baselines outperform the GloVe based baselines. The reason is that BERT learns context-dependent word embeddings, which could better reflect the meaning of a word in specific context and lead to better performance for downstream tasks.

2) Comparison with the Joint Learning Baselines: Tables VIII and IX report the results where we compare Joint-WB with other joint models and variants of Joint-WB. Overall, Joint-WB outperforms the best baseline by 0.12% in F1 for attribute extraction and by 0.29% in EM for topic generation. This is because of the signal exchange and enhancement mechanisms used in our model, which improves model's capability to utilize the hints from other sub-tasks.

The concat-based signal exchange methods are slightly better than Naive-Join for attribute extraction, and are the same as Naive-Join for topic generation. The slight improvements are expected since all the words are regarded equally towards a topic, while actually different words in a webpage should gain different attention with a given topic. In comparison, the attention-based signal exchange yield more improvements for both tasks than concat-based methods, i.e., up to 1.96% in F1 for attribute extraction and 0.49% in EM for topic generation.

TABLE VIII
RESULTS OF DIFFERENT JOINT MODELS FOR ATTRIBUTE EXTRACTION.

Methods		P	R	F1
No signal exchange	Naive-Join	96.27	93.14	94.68
Concat-based signal exchange	Con-Extractor	96.38	93.72	95.03
	Ave-Extractor	96.48	93.69	95.07
Attention-based signal exchange	Att-Extractor	97.73	94.02	95.84
	Att-Extractor+Att-Generator	98.20	95.81	96.99
Section signal enhancement	Pip-Extractor+Pip-Generator	98.31	96.07	97.18
	Joint-WB (our proposed)	**98.42**	**96.21**	**97.30**

TABLE IX
RESULTS OF DIFFERENT JOINT MODELS FOR TOPIC GENERATION.

Methods		EM	RM
No signal exchange	Naive-Join	93.70	95.11
Concat-based signal exchange	Con-Extractor	93.71	95.11
	Ave-Extractor	93.71	95.16
Attention-based signal exchange	Att-Extractor	93.82	95.20
	Att-Extractor+Att-Generator	94.20	96.31
Section signal enhancement	Pip-Extractor+Pip-Generator	94.74	96.85
	Joint-WB (our proposed)	**95.02**	**97.24**

Enhancing the signal from the informative section prediction task using pipeline methods (Pip-Extractor+Pip-Generator) also contribute to the model performance. The improvement trend is consistent with the observation on adding prior knowledge about informative sections to single-task models. Joint-WB model outperforms the pipeline based models since dual-aware extractor or generator learn lead to more universal hidden representations than pipeline extractor or generator.

D. Model Sensitivity

We inspect the content sensitivity of the proposed Joint-WB model, and the proposed Dual-Distill and Tri-Distill methods with Joint-WB as the teacher on 300 synthetic webpages. We concate the content of two real webpages with different topics into a synthetic webpage. The proportion of content length of the two real webpages is controlled to 50%-50%, 70%-30% and 30%-70%. We observe that the Joint-WB model without any distillation always predicts based on the content appear first in the synthetic webpage, while the Dual-Distill and Tri-Distill methods tend to predict based on the content with a larger portion. This indicates that Joint-WB is more sensitive to the content position while the Dual-Distill and Tri-Distill are more sensitive to the content length. In future work, we plan to explore WB models which provide more hierarchical summary for webpages with combination contents.

E. Human Evaluation

To complement the automatic evaluation for topic generation, we also perform a human evaluation on 40 randomly

TABLE X
AVERAGE SCORE OF HUMAN EVALUATION OF DIFFERENT MODELS FOR TOPIC GENERATION.

Methods	Seen domians	Unseen domians
BERT↦[Bi-LSTM,LSTM]	1.30	0.97
BERTSUM↦[Bi-LSTM,LSTM]	1.35	0.99
Naive joint	1.49	1.08
Att-Extractor + Att-Generator	1.60	1.20
Pip-Extractor + Pip-Generator	1.64	1.23
ID only	1.78	1.71
UD only	1.75	1.74
Tri-Distill (our proposed)	**1.83**	**1.81**
Full score	2.00	2.00

selected webpages with previous unseen topics, and 40 randomly selected webpages with previous seen topics. We ask ten volunteers to assign a score of 2 (perfectly suitable), 1 (suitable) or 0 (unsuitable) to each webpage to indicate whether the generated topics from different models are suitable for the webpage. All the volunteers have studied English for at least ten years and are trained on the scoring criteria with ten examples for 25 minutes. We compute the inter-annotator agreement using Cohen's κ measurement. The result shows that volunteers have high agreement (κ >0.83) for generated topics from different models on different webpages. Table X shows the average score of the human evaluations, which is consistent with the results in automatic evaluation. Our proposed models achieves better scores than different baselines on webpages with previously unseen domains.

V. Conclusion

We proposed the new task of webpage briefing (WB), which provides a summary of a webpage in a hierarchical manner to help increase the speed of webpage browsing. We propose three models for the task, Dual-Distill, Tri-Distill, and Joint-WB. Dual-Distill consists of identification distillation and understanding distillation, where the former distills knowledge on identifying informative contents under the guide of predefined topics, while the later distills knowledge on topic generation or key attribute extraction. Tri-Distill consists of a shared identification distillation and two understanding distillations, one for topic generation and the other for key attribute extraction. Joint-WB has a joint learning architecture with signal exchange and enhancement mechanisms among a key attribute extractor, a topic generator, and an informative section predictor. Experiments using real-world webpages show that Dual-Distill and Tri-Distill achieves high performance for WB with 94.86% in EM and 58.20% in F1 on unseen domains. They outperform baselines in different settings by at most 7.03% in F1 and 8.63% in EM. Experimental results also show that Joint-WB achieves 97.30% in F1 and 95.02% in EM on seen domains. It outperforms single-task baselines by at most 7.73% in F1 and 9.65% in EM, and also outperforms other jointly trained baselines. Human evaluation validates the effectiveness of the proposed models.

For future work, we aim to extend the proposed models and experimental study to more levels of hierarchy. We also plan to predict attribute names for key attributes (e.g., in Fig. 1, the attribute name for the key attribute '*$40.13*' is '*Price*').

Acknowledgment

This work is supported by Australian Research Council Discovery Project DP180102050.

References

[1] N. Akhtar, B. Siddique, and R. Afroz, "Visual and textual summarization of webpages," in *ICDMIC*, 2014.

[2] A. L. Berger and V. O. Mittal, "Ocelot: a system for summarizing web pages," in *SIGIR*, 2000.

[3] J.-Y. Delort, B. Bouchon-Meunier, and M. Rifqi, "Enhanced web document summarization using hyperlinks," in *HT*, 2003.

[4] M.-T. Nguyen, D.-V. Tran, L.-M. Nguyen, and X.-H. Phan, "Exploiting user posts for web document summarization," *TKDD*, vol. 12, no. 4, pp. 1–28, 2018.

[5] T. Manabe and K. Tajima, "Extracting logical hierarchical structure of html documents based on headings," *PVLDB*, vol. 8, no. 12, pp. 1606–1617, 2015.

[6] H. Saggion and T. Poibeau, "Automatic text summarization: Past, present and future," in *Multi-source, multilingual information extraction and summarization*, 2013, pp. 3–21.

[7] S. Teufel and M. Moens, "Summarizing scientific articles: experiments with relevance and rhetorical status," *Computational linguistics*, vol. 28, no. 4, pp. 409–445, 2002.

[8] R. Nallapati, B. Zhou, C. dos Santos, Ç. glar Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," 2016.

[9] L. Xiong, C. Hu, C. Xiong, D. Campos, and A. Overwijk, "Open domain web keyphrase extraction beyond language modeling," in *EMNLP-IJCNLP*, 2019.

[10] S. Sun, C. Xiong, Z. Liu, Z. Liu, and J. Bao, "Joint keyphrase chunking and salience ranking with bert," *arXiv:2004.13639*, 2020.

[11] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," in *ACL*, 2017.

[12] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *KDD*, 2006.

[13] P. Micaelli and A. J. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," in *NeurIPS*, 2019.

[14] G. Menghani and S. Ravi, "Learning from a teacher using unlabeled data," *arXiv:1911.05275*, 2019.

[15] Y. Hao, Y. Fu, Y.-G. Jiang, and Q. Tian, "An end-to-end architecture for class-incremental object detection with knowledge distillation," in *ICME*, 2019.

[16] U. Michieli and P. Zanuttigh, "Knowledge distillation for incremental learning in semantic segmentation," *arXiv:1911.03462*, 2019.

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[18] D. Ma, S. Li, and H. Wang, "Joint learning for targeted sentiment analysis," in *EMNLP*, 2018.

[19] Y. Dai, R. Zhang, and J. Qi, "Person name recognition with fine-grained annotation," in *JCDL*, 2020.

[20] Y. Dai, J. Qi, and R. Zhang, "Joint recognition of names and publications in academic homepages," in *WSDMs*, 2020.

[21] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *EMNLP-IJCNLP*, 2019.

[22] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv:1508.01991*, 2015.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.

[24] M. L. Vidal, A. S. da Silva, E. S. de Moura, and J. M. Cavalcanti, "Structure-driven crawler generation by example," in *SIGIR*, 2006.

[25] Q. Hao, R. Cai, Y. Pang, and L. Zhang, "From one tree to a forest: a unified solution for structured web data extraction," in *SIGIR*, 2011.

[26] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[28] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *EMNLP*, 2016.