Continuous Prompt Tuning Based Textual Entailment Model for E-commerce Entity Typing

Yibo Wang
Department of Computer Science
University of Illinois at Chicago
Chicago, IL, USA
ywang633@uic.edu

Congying Xia
Saleforce Research
Palo Alto, CA, USA
c.xia@salesforce.com

Guan Wang
Huski Inc
San Jose, CA, USA
guan.w@huski.ai

Philip S. Yu
Department of Computer Science
University of Illinois at Chicago
Chicago, IL, USA
psyu@uic.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020766

***Abstract*—The explosion of e-commerce has caused the need for processing and analysis of product titles, like entity typing in product titles. However, the rapid activity in e-commerce has led to the rapid emergence of new entities, which is difficult for general entity typing. Besides, product titles in e-commerce have very different language styles from text data in general domain. In order to handle new entities in product titles and address the special language styles of product titles in e-commerce domain, we propose our textual entailment model with continuous prompt tuning based hypotheses and fusion embeddings for e-commerce entity typing. First, we reformulate entity typing into a textual entailment problem to handle new entities that are not present during training. Second, we design a model to automatically generate textual entailment hypotheses using a continuous prompt tuning method, which can generate better textual entailment hypotheses without manual design. Third, we utilize the fusion embeddings of BERT embedding and CharacterBERT embedding to solve the problem that the language styles of product titles in e-commerce are different from that of general domain. To analyze the effect of each contribution, we compare the performance of entity typing and textual entailment model, and conduct ablation studies on continuous prompt tuning and fusion embeddings. We also evaluate the impact of different prompt template initialization for the continuous prompt tuning. We show our proposed model improves the average F1 score by around 2% compared to the baseline BERT entity typing model.**

***Index Terms*—continuous prompt tuning, textual entailment model, characterBERT, e-commerce entity typing**

I. Introduction

Nowadays, the boom of e-commerce has led to an increasing preference for online shopping. In order to better manage products and provide services to customers, such as classifying products and recommending products, e-commerce platforms need to understand entities in the product titles, such as brand names (Apple, Nike, etc.), product names (iPhone, shoes, etc.) and other features (colors, sizes, etc.). This task is usually formulated as entity typing, which is to classify entity types given the entity and context. For example, a product title 'NAGANO Set of 2 Chairs' in Table I has 'NAGANO' as a brand name and 'Chairs' as a product name. Thus the entity typing task is to classify 'NAGANO' and 'Chairs' into 'brand name', 'product name' or 'feature' based on the product title.

TABLE I: Sample data.

Product Titles*
NAGANO Set of 2 **Chairs**
Maison Louis Marie **Perfume Oil Discovery Set**
Hugo Boss (Dark Blue for Men 2.5 Oz) **EDT**
(100% Cotton) Unpaper **Towels** - *Cheeks Ahoy*

*Italics indicates brand; Bold indicates product; Bracket indicates feature.

Traditional approaches treat entity typing as text classification with a pre-defined label set. However, new brands, products and features are emerging all the time on e-commerce platforms. Previous entity typing methods lack the flexibility to welcome new entities and require a large amount of training data to obtain decent performance. Thus, in this paper we propose a novel approach to formulate entity typing as a textual entailment problem. This textual entailment formulation gives our model the ability to easily adapt to new entities [1]. A textual entailment model is to determine whether two text fragments have a relationship that one fragment implies the other, that is, whether the two text fragments are entailment or not [2]. In our work, we formulate the textual entailment model by using the product title as one fragment of text and hypotheses with unfilled slots for entities to be classified as the other fragment of text. So the textual entailment task is to determine whether the product title and the hypotheses with entities to be classified are entailment or not. For example, when the product title is 'Nike Mens Sportswear Aimoji Hoodie', the entity to be classified is 'Hoodie' and entity labels are directly used as hypotheses, the textual entailment model input will be 'Nike Mens Sportswear Aimoji Hoodie [SEP] Hoodie is a brand name/ product/ feature', where 'is a brand name/ product/ feature' is the hypotheses.

TABLE II: Examples of human designed hypotheses for our textual entailment models.

Labels	Example Hypotheses Using Label Names	Example Hypotheses Using Label Dictionary Explanations
Brand	'entity' is a brand name	'entity' is a band name, which is a type of things manufactured by a particular company under a particular name
Product	'entity' is a product	'entity' is a product, which is an article or substance that is manufactured or refined for sale
Feature	'entity' is a feature	'entity' is a feature, which is a distinctive attribute or aspect of something

Designing the hypotheses is the most essential part in the textual entailment task [1]. Normally people manually design multiple templates for experiments, and then choose the templates with best experimental performance. Previous works usually convert labels or dictionary explanations into hypotheses with human-designed templates as shown in Table II. However, the model performance cannot be guaranteed to be satisfactory using human designed hypotheses. Not to mention that it requires a lot of human effort and domain knowledge to design and test these templates. Therefore, we propose a special continuous prompt tuning method to automatically find the best hypotheses for textual entailment models. Prompt tuning is the process of creating prompt templates (hypotheses in our case) that result in the most effective performance on the downstream tasks, and continuous prompt tuning is to directly create prompt templates in the embedding space rather than creating prompt templates that human can understand [3].

Besides, product title data has its own language styles which are different from text data in general domain. Most product titles are not complete sentences, but phrases or a bunch of keywords which summarize and describe products. For example, 'Trendy Apple Sports iWatch Band' is a phrase and 'Zebra Sarasa Retractable Gel Pen — Blue — Medium 0.7 mm' is a set of keywords. Also, many special abbreviations like 'Pcs' in 'INGCO 142 Pcs combination tools set', self-made words like 'Fashern' in 'Fashern snake skin print rebecca set' and translated words like 'GaGaZui' in 'GaGaZui Green Bean Sweet Spicy Flavor 33g * 30 bags 990g' are in product titles. In order to adjust to these e-commerce domain special language styles, we utilize fusion embeddings of BERT and CharacterBERT [4], which is a character-level language model that includes a Character-CNN module to represent entire words by deliberating their characters rather than using predefined word-piece vocabularies from general domain as in BERT [5]. By using fusion embeddings of BERT embedding and CharacterBERT embedding, the special language styles of product titles and the tokens that do not appear in the predefined word-piece vocabularies can be handled better.

In this paper, we propose **C**ontinuous Prompt Tuning based **T**extual Entailment **M**odel (CTM) to handle new entities in product titles, automatically generate optimized textual entailment hypotheses and address the special language styles problem of e-commerce domain. Our primary contributions include:

- Formulating entity typing in product titles into a novel textual entailment problem and building the textual entailment model hypotheses using continuous prompt tuning methods: The textual entailment formulation provides model with the ability to handle new entities that are not present in training set.
- Building a continuous prompt tuning model for BERT and characterBERT: The continuous prompt tuning model saves human effort to design textual entailment hypotheses and is able to automatically build better hypotheses.
- Utilizing the fusion embeddings of BERT embedding and CharacterBERT embedding: The fusion embeddings of BERT embedding and CharacterBERT embedding allow our proposed CTM model to handle the special language styles of e-commerce product titles.

A more detailed version can be found at [18]. Our codes are available at https://github.com/YiboWANG214/CTM.

II. Related Work

Textual entailment problem is a frequently studied task, and both prompt learning and character-level language models are very popular in recent years.

Recently, lots of work has been studied for converting a classification task into a textual entailment problem. For example, [6] converts a relation classification task into a textual entailment problem, where the hypotheses are relation descriptions. [1] treats the zero shot text classification task as a textual entailment problem, so that their model can achieve knowledge from other entailment datasets. The textual entailment problem has great potential because the changes and design of hypotheses can bring about many model modifications and improvements.

Since the publication and popularity of GPT-3 [7], prompt learning has been payed great attention to. As mentioned in [3], prompt tuning is the phase of creating effective prompt templates for downstream tasks, which is one of the most important phases of prompt learning. Prompt tuning can be classified into hand-crafted prompt tuning and automated prompt tuning. Many famous work has exploited hand-crafted prompt tuning, like GPT-3 [7], T5 [8] , etc. Manually designed prompt templates do not require computational resources and work well in many cases. However, exploring optimal prompt templates via hand-crafted prompt tuning is hard and requires time and domain knowledge. Thus, automated prompt tuning [9]–[11] is proposed to solve these problems. Prompt learning has an effect comparable to or even better than fine-tuning on large language models such as GPT-3 and T5 [10], [11] and smaller language models like [12]. The success of prompt learning inspires us to apply the idea of continuous prompt tuning to design textual entailment hypotheses for entity typing task.

Many character-level language models based on different methods have been proposed to tackle the problem caused by subword-level models. [13] introduces a hierarchical RNN based character-level language model. [14] proposes a model to apply a CNN and a highway network over characters, whose output is sent to a LSTM lanaguage model. CharacterBERT [4] utilizes a Character-CNN module to represent tokens by their characters; CharBERT [15] uses context string embeddings and a heterogeneous interaction module to obtain character representations; CharFormer [16] use a soft gradient-based subword tokenization module to learn subword representation from characters. Canine [17] is a neural encoder that operates directly on character sequences without subword tokenization or vocabulary.

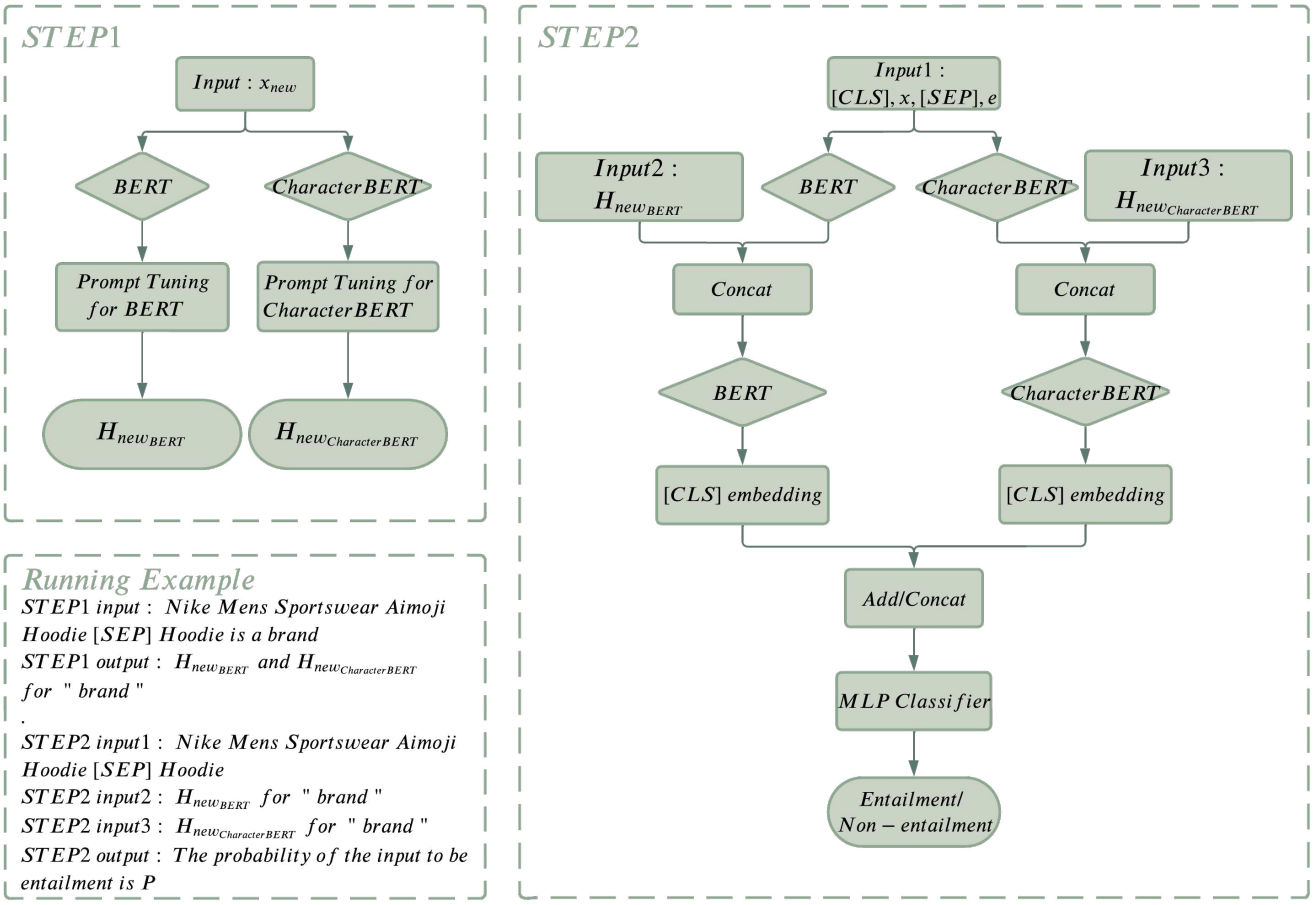

Fig. 1: Model architecture. STEP 1 is the process of obtaining optimized hypotheses by applying prompt tuning to BERT and CharacterBERT separately, and STEP 2 is the process of using textual entailment model and fusion embedding for classification. A running example is shown in the figure.

III. Methods

In this work, we reformulate the entity typing task in product titles into a textual entailment problem. To save human effort and improve model performance, we utilize continuous prompt tuning to automatically create optimized hypotheses for the textual entailment model. Additionally, we fuse BERT embedding and characterBERT embedding to acclimate to special language styles of product titles.

A. Problem Formulation

Given a product title x and an entity e which is a substring of x, the original entity typing task is to classify the entity e into an entity type $y \in \{$'brand', 'product', 'feature'$\}$ for each input product title x.

In order to format the entity typing task into a textual entailment problem, we concatenate the product title x, the entity e and the hypothesis h for different classes together and send them into the textual entailment model. The new input to the textual entailment model is formulated as follows:

$$x_{new} = [CLS], x, [SEP], e, h, \tag{1}$$

where $h \in \mathbb{H}$, and $\mathbb{H}$ is the hypotheses set with one hypothesis for each class. The objective of the textual entailment model is to predict whether the two text fragments of the concatenated new input is entailment or not, which is to classify the new input x_{new} into a binary class $y_{new} \in \{$'entailment', 'non-entailment'$\}$.

B. Textual Entailment model

In order to train a textual entailment model, we need to construct positive/negative entailment pairs. For the positive pairs, we concatenate the sentence-entity pair and the hypothesis from its ground truth label as the positive entailment example. For negative pairs, we randomly select one hypothesis from negative labels to construct 'non-entailment' example.

During training, we train the textual entailment model with the cross entropy loss on the constructed positive and negative entailment pairs. For the inference, we concatenate each test example with hypothesis from different classes and we choose the class with the highest probability as the predicted entity type.

C. Continuous Prompt Tuning

Inspired by soft prompt tuning [11], we design a novel continuous prompt tuning method to obtain optimized hypotheses for the textual entailment task.

Given a product title with n tokens $x = \{x_1, \ldots, x_n\}$ and the corresponding entity with m tokens $e = \{e_1, \ldots, e_m\}$, the embedding matrix $X_e = [X; E] \in \mathbb{R}^{(n+m)\times d}$ is obtained by the language model, where $X \in \mathbb{R}^{n\times d}$ is the embedding matrix of the product title x, $E \in \mathbb{R}^{m\times d}$ is the embedding matrix of the entity e and d is the dimension of the embedding space. The continuous prompt template (hypothesis) h is represented as $H \in \mathbb{R}^{p\times d}$, where p is the length of the prompt tokens. Then the embedding matrix X_e and the prompt matrix H are concatenated as $[X_e; H] \in \mathbb{R}^{(n+m+p)\times d}$ to be the input embedding of the language model. The prompt tuning phase is then modeled as a masked language modeling. During the fine-tuning of the masked language modeling, the parameters θ of the language model are frozen, and the only tunable parameters are the embedding matrix H of the hypothesis. For each class, the optimized embedding matrix $H_{new} \in \mathbb{R}^{p\times d}$ of one hypothesis is generated by using all training data in this class. Later these embedding matrices will be used as the optimized hypotheses in the textual entailment model.

The initialization of hypotheses is critical to the continuous prompt tuning and the textual entailment performance. We use different lengths and different types of hypotheses initialization. The first type of hypotheses is directly using class labels to formulate the initialized hypotheses. The length of the hypotheses is 3 and the hypotheses are 'is a brand/ is a product/ is a feature'. The second type of hypotheses is converted from dictionary explanations of the class labels. The length of the hypotheses is 14 and the initialized hypotheses are 'is a brand, which is a type of things manufactured by a particular company/ is a product, which is an article or substance manufactured or refined for sale/ is a feature, which is a distinctive attribute or a special aspect of something'. Different hypotheses initialization leads to different performance.

D. Fusion Embeddings

For CharacterBERT, each token is converted into a sequence of characters and then represented as a 50-dimension input tensor. Then a Character-CNN module will be used to project the 50-dimension input tensor into a 768-dimension CharacterBERT embedding which is selected to be dimensionally aligned with BERT.

The optimized hypothesis using BERT embedding $H_{new_{BERT}}$ and the optimized hypothesis using CharacterBERT embedding $H_{new_{CharacterBERT}}$ for each class can be obtained by applying continuous prompt tuning separately on BERT and CharacterBERT. Then $H_{new_{BERT}}$ is concatenated to the BERT embedding of the product title and the entity, and the concatenated new BERT embedding is fed into the BERT model for a better contextual embedding. Similarly, $H_{new_{CharacterBERT}}$ is concatenated to the CharacterBERT embedding of the product title and the entity, and the concatenated new CharacterBERT embedding is fed into the CharacterBERT model.

After obtaining the better contextual BERT embedding and CharacterBERT embedding, we use different fusion methods to obtain the fusion embedding of BERT embedding and CharacterBERT embedding. The same embedding dimension of CharacterBERT and BERT allows diverse fusion methods to be tried. We tried two simple but effective fusion methods: 1). concatenating '[CLS]' BERT embedding (V_{BERT}) and weighted '[CLS]' CharacterBERT embedding ($V_{CharacterBERT}$), and 2). adding '[CLS]' BERT embedding (V_{BERT}) to weighed '[CLS]' CharacterBERT embedding ($V_{CharacterBERT}$). We use '[CLS]' embedding because '[CLS]' embedding is considered to accommodate sentence information.

For the first fusion method, the fusion embedding for one sample is

$$[V_{BERT} \quad \alpha * V_{CharacterBERT}], \tag{2}$$

where V_{BERT} is '[CLS]' BERT embedding, $V_{CharacterBERT}$ is '[CLS]' CharacterBERT embedding and α is a tunable weight parameter. The fusion embedding shape for a batch is [batch size, 2*hidden_size]. For the second method, the fusion embedding for one sample is

$$V_{BERT} + \alpha * V_{CharacterBERT}. \tag{3}$$

The fusion embedding shape for a batch is [batch size, hidden_size]. The fusion embeddings of input are then fed into two-layer MLPs for textual entailment classification.

In summary, as in Fig. 1, the input of STEP 1 is x_{new} in (1), which is the new input for the textual entailment formulation and the output of STEP 1 is the optimized hypothesis using BERT embedding $H_{new_{BERT}}$ and the optimized hypothesis using CharacterBERT embedding $H_{new_{CharacterBERT}}$. The input of STEP 2 is x, e, $H_{new_{BERT}}$ and $H_{new_{CharacterBERT}}$, and the output of STEP 2 is the probability of the input to be entailment.

IV. Experiments and Results

In this section, we compare the performance of our proposed CTM model and the baseline models on a product title dataset. Then we compare model performance of entity typing and textual entailment model on test set and novel entity set and conduct a series of ablation studies to investigate the effect of each contribution.

A. Dataset

The raw data is provided by Huski.ai . We randomly select 10,000 English product titles from the raw data, and hire three workers to label each token in product titles as one of the brand name, product/ product name, feature or others. We then filter the data based on the majority and manually double check the data. After filtering, the training set has 2324 samples in total with 797 'brand', 700 'product' and 827 'feature', and the test set has 1317 samples in total with 360 'brand', 437 'product'

https://huski.ai

TABLE III: F1 score comparison between our proposed CTM models and the baseline model.

Hypotheses / Initialization	Model	Brand	Product	Feature	Average
-	Entity Typing	0.8602	0.8391	0.8754	0.8595
Class Labels	CTM (add)	**0.8809**	0.8555	0.8931	0.8770
	CTM (concate)	0.8776	0.8587	0.8871	0.8747
Dictionary Explanations	CTM (add)	0.8760	0.8636	0.8906	**0.8778**
	CTM (concate)	0.8679	**0.8666**	**0.8946**	**0.8778**

TABLE IV: Class F1 scores and Average F1 scores for entity typing and textual entailment model with labels / dictionary explanations as hypotheses on test set and novel entity set.

Dataset	Model	Brand	Product	Feature	Average
Test Set	Entity Typing	0.8602	0.8391	**0.8754**	0.8595
	Textual Entailment (label)	0.8568	0.8395	0.8605	0.8572
	Textual Entailment (dictionary)	**0.8776**	**0.8451**	0.8727	**0.8648**
Novel Entity Set	Entity Typing	0.8925	0.7860	0.8632	0.8601
	Textual Entailment (label)	0.9094	0.7833	0.8693	0.8689
	Textual Entailment (dictionary)	**0.9139**	**0.7985**	**0.8772**	**0.8759**

and 520 'feature'. We also create a novel entity set, which is filtered from the test set, only with new entities that are not present in training set. The novel entity set has 572 samples in total with 267 'brand', 135 'product' and 170 'feature'. This novel entity set is used to compare the ability to handle new entities between the entity typing task and the textual entailment formulation. Sample data is shown in Table I.

B. Experimental Results and Analysis

Table III exhibits the F1 score comparison between our proposed CTM models with different fusion methods and prompt initialization and the baseline entity typing model. Any version of our CTM model has better performance compared to the entity typing model, and CTM with concatenation as fusion method and dictionary explanations as prompt initialization has the best performance in general.

To analyze the contributions of different components in our proposed CTM model, we compare model performance of entity typing and textual entailment model on test set and novel entity set, and perform ablation studies on prompt tuning and fusion embedding on test set.

TABLE V: Class F1 scores and Average F1 scores for CTM and CTM without prompt tuning (PT) to evaluate the contribution of prompt tuning. All the results are evaluated on the test set.

Hypotheses / Initialization	Model	Brand	Product	Feature	Average
Class Labels	CTM (add)	**0.8809**	0.8555	**0.8931**	**0.8770**
	(w/o) PT	0.8808	**0.8575**	0.8836	0.8740
	CTM (concate)	0.8776	**0.8587**	**0.8871**	**0.8747**
	(w/o) PT	**0.8862**	0.8532	0.8749	0.8709
Dictionary Explanations	CTM (add)	**0.8760**	**0.8636**	**0.8906**	**0.8778**
	(w/o) PT	0.8717	0.8523	0.8787	0.8679
	CTM (concate)	**0.8679**	**0.8666**	**0.8946**	**0.8778**
	(w/o) PT	0.8668	0.8477	0.8784	0.8648

TABLE VI: Class F1 scores and Average F1 scores for CTM and CTM without fusion embedding to evaluate the contribution of fusion embedding. All the results are evaluated on the test set.

Hypotheses / Initialization	Model	Brand	Product	Feature	Average
Class Labels	CTM (add)	**0.8809**	0.8555	**0.8931**	**0.8770**
	CTM (concate)	0.8776	**0.8587**	0.8871	0.8747
	CTM (CharacterBERT)	0.8683	0.8459	0.8751	0.8633
	CTM (BERT)	0.8607	0.8368	0.8799	0.8610
Dictionary Explanations	CTM (add)	0.8760	0.8636	0.8906	**0.8778**
	CTM (concate)	0.8679	**0.8666**	**0.8946**	**0.8778**
	CTM (CharacterBERT)	0.8667	0.8389	0.8776	0.8618
	CTM (BERT)	**0.8850**	0.8492	0.8809	0.8717

Textual Entailment Formulation In order to compare the ability to handle new entities between entity typing formulation and textual entailment formulation, we conduct comparative experiments between entity typing and textual entailment on both test set and novel entity set. We also conduct experiments on textual entailment model with different handcrafted hypotheses to verify impact of different hypotheses. The regular entity typing model is used as baseline, and BERT is used as a backbone model for both entity typing and textual entailment model. The results are shown in Table IV. For test set, the difference of performance between entity typing and textual entailment is relatively small and the average F1 score of entity typing is even slightly better than textual entailment model with labels as hypotheses. For novel entity set, the performance improvement obtained by textual entailment formulation is more significant for hypotheses converted from both labels and dictionary explanations. For both sets, the average F1 scores of the textual entailment model with dictionary explanations as hypotheses are higher than that of the textual entailment model with labels as hypotheses. The more obvious improvement of textual entailment model on novel entity set compared to the improvement of textual entailment model on test set shows that the textual entailment formulation provides model with the ability to better handle new entities. Beside, the format of textual entailment model provides more possibilities for performance improvement, like the selection of hypotheses. The performance difference between two hypotheses indicates that different textual entailment hypotheses lead to different model performance, which confirms our previous thinking that model performance cannot always be guaranteed by human-designed hypotheses and indicates the importance of the selection of hypotheses.

Prompt Tuning In order to evaluate the contribution of prompt tuning and verify whether the hypotheses obtained by prompt tuning work better than hand-crafted hypotheses, we compare the performance between the CTM models and the similar models without prompt tuning based hypotheses. Besides, to explore the effect of different prompt initialization for prompt tuning, the experiments are performed on hypotheses and prompt initialization converted both from labels and dictionary explanations. The results are shown in Table V. Whether using

class labels or dictionary explanations as prompt initialization, the hypotheses obtained by prompt tuning perform better than the hypotheses converted manually from class labels or dictionary explanations, which implies that prompt tuning is effective for model improvement. When applying hypotheses converted from class labels, the model improvement obtained by using prompt tuning is small; when applying hypotheses converted from dictionary explanations, the model improvement achieved by using prompt tuning is greater. One possible explanation is that the dictionary explanations induce longer hypotheses, allowing more parameters to be tuned during the prompt tuning phase. A good prompt template initialization can lead to better model performance.

Fusion Embedding In order to compare the performance between fusion embedding and BERT / CharacterBERT embedding, we conduct experiments on the CTM models and the similar models using BERT embeddings or CharacterBERT embeddings alone on hypotheses and prompt initialization converted from both labels and dictionary explanations. In addition, we build the same architecture for BERT embedding and CharacterBERT embedding, which is using a two-layer MLP classifier for embedding classification, for better comparison. As shown in Table VI, fusion embedding works much better than BERT embedding or CharacterBERT embedding either using class labels or dictionary explanations as hypotheses and prompt initialization. This is expected because by using fusion embedding, we cover both word-piece-level information and character-level information, which makes the model more adaptable to product title language style and special e-commerce product title vocabulary.

V. Conclusions

We reformulate entity typing in product titles into a novel textual entailment model and utilize the continuous prompt tuning to automatically create entailment hypotheses. The textual entailment formulation provides model with the ability to better handle new entities that are not present in the training set. And the use of the continuous prompt tuning eliminates the trouble of manually designing hypotheses and creates better hypotheses to increase the model performance. We try different types of prompt templates to initialize the continuous prompt tuning method and analyze the different results obtained. Besides, we apply the fusion embedding of BERT embedding and CharacterBERT embedding to address the different language styles of product titles and special vocabularies in e-commerce. In conclusion, our proposed CTM model has the ability to deal with new entities and the special language styles of product titles.

Acknowledgment

This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

Thank Huski.ai for sponsorship and data support.

References

[1] W. Yin, J. Hay, and D. Roth, "Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach." In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3914-3923. 2019.

[2] D. Z. Korman, E. Mack, J. Jett, and A. H. Renear. "Defining textual entailment." Journal of the Association for Information Science Technology 69, no. 6 (2018): 763-772.

[3] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." arXiv preprint arXiv:2107.13586 (2021).

[4] H. E. Boukkouri, O. Ferret, T. Lavergne, H. Noji, P. Zweigenbaum, and J. Tsujii, "CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters." In Proceedings of the 28th International Conference on Computational Linguistics, pp. 6903-6915. 2020.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "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, Volume 1 (Long and Short Papers), pp. 4171-4186. 2019.

[6] A. Obamuyide, and A. Vlachos, "Zero-shot relation classification as textual entailment." In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pp. 72-78. 2018.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan et al, "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer." J. Mach. Learn. Res. 21, no. 140 (2020): 1-67.

[9] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts." In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4222-4235. 2020.

[10] X. L. Li, and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation." In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4582-4597. 2021.

[11] B. Lester, R. Al-Rfou, and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning." In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3045-3059. 2021.

[12] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang, "P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks." In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 61-68. 2022.

[13] K. Hwang, and W. Sung, "Character-level language modeling with hierarchical recurrent neural networks." In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5720-5724. IEEE, 2017.

[14] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models." In Thirtieth AAAI conference on artificial intelligence. 2016.

[15] W. Ma, Y. Cui, C. Si, T. Liu, S. Wang, and G. Hu, "CharBERT: Character-aware Pre-trained Language Model." In Proceedings of the 28th International Conference on Computational Linguistics, pp. 39-50. 2020.

[16] Y. Tay, V. Q. Tran, S. Ruder, J. Gupta, H. W. Chung, D. Bahri, Z. Qin, S. Baumgartner, C. Yu, and D. Metzler, "Charformer: Fast character transformers via gradient-based subword tokenization." arXiv preprint arXiv:2106.12672 (2021).

[17] J. H. Clark, D. Garrette, I. Turc, and J. Wieting, "Canine: Pre-training an efficient tokenization-free encoder for language representation." Transactions of the Association for Computational Linguistics 10 (2022): 73-91.

[18] Y. Wang, C. Xia, G. Wang, and P. S. Yu, "Continuous Prompt Tuning Based Textual Entailment Model for E-commerce Entity Typing." arXiv preprint arXiv:2211.02483 (2022).