Addressing Idiom Identification from Generative Perspective

Anonymous ACL submission

Abstract

Idiomatic expressions are a vital part of natural language, and new ones are constantly being added. The general token-level sequence labeling method for identifying idiomatic expressions has potential limitations due to the semantic non-compositionality of idioms, as it is challenging for the model to capture the integrated semantic of multi-word expressions. We present Idiom-T5 that leverages a pre-trained Transformer-based model to generate idiomatic expression from the source sentences. We observe that Idiom-T5 has powerful generalization to unknown idioms, outperforming BERT by 14% on sequence accuracy in zero-shot setting. Furthermore, Idiom-T5 performs well in data-scarce scenarios, achieving 96% accuracy with only 6% of the data. We then propose a simple but effective data augmentation method to improve the performance of Idiom-T5 in data-scarce scenarios.

1 Introduction

Idiomatic expression (IEs) are a special class of multi-word expression (MWEs) that typically occur as collocations and exhibit semantic non-compositionality, where the meaning of expression may not be directly related to the meanings of their individual words (e.g., big fish as an important person), and also sometimes depend on its context (e.g. behind someone back as stealthily) (Baldwin and Kim, 2010). Following the terminology from (Haagsma et al., 2020; Zeng and Bhat, 2021), we also call these phrase as potentially idiomatic expressions (PIEs) to account for the contextual semantic ambiguity. This poses the challenge for current NLP applications, as (Yu and Ettinger, 2020) show that phrase representation in the pre-trained language models relies heavily on word content, with little evidence of nuanced composition. One of the task to explore the problem is the idiom expression identification, either can be on sequence-level or token-level, which is the MWE identification problem defined by Baldwin and Kim (2010) limited to MWEs with semantic idiomaticity.

(Zeng and Bhat, 2021) treated the idiomatic expression identification as a sequence labeling problem, and proposed a multi-stage neural architecture with attention flow, which effectively fuses contextual and lexical information at different levels using word and sub-word representations based on the semantic compatibility (Liu and Hwa, 2019) of idiom expressions. However, recent studies (Yu and Ettinger, 2020) found that the phrase representation obtained from state-of-the-art transformers (e.g. BERT (Devlin et al., 2019a), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019)), fail to reflect sophisticated phrase composition beyond what can be gleaned from word content, even the models fine-tuned by downstream tasks (Yu and Ettinger, 2021). (Garcia et al., 2021) also indicated that idiomaticity is not yet accurately represented by contextualised models. Therefore, we argue that the sequence labeling approach for idiom expression identification might be sub-optimal, since the token-level training objective does not allow the model to learn phrase-level semantic compatibility better. Alternatively, the span-based approach (Kwiatkowski et al., 2019) is more suitable for this task.

In this work, we formulate the idiom expression identification as a generation task and utilize the pre-trained generative language model T5 (Raffel et al., 2020), which means we attempt to directly
generate the target idiom expression appearing in
the source sentence. This generative training objec-
tive is much stricter and more challenging, but may
focus the model to pay more attention to the phrase-
level semantic representation. Another benefit is
the pre-training objective of T5 is "recover cor-
rupted spans", hence it can be more sensitive with
the semantic continuity of phrase. Furthermore,
we find our generative approach performs well
in data-scarce scenarios. While data augmentation
is an effective technique used in smaller datasets
for text classification tasks(Wei and Zou, 2019; Xie
et al., 2020), we also propose a simple data augmen-
tation method for our generative idiom expression
identification task, to improve the performance of
T5 in data-scarce scenarios.

2 Approach

In this section, we introduce sequence labeling and
span-based models, followed by a description of
our proposed generative method, as shown in Fig-
ure 1. And finally introduce our data augmentation
approach.

2.1 preliminaries

The task formulation is: given an input sequence
X = {x1, x2, ..., xn}, where n denotes the length of
the sequence, we need to find every PIEs in X,
and assign a label y ∈ Y to every x, where Y is
{literal, idiom}. For training, we first encode the
sentence with a pre-trained language model, such as
BERT(Devlin et al., 2019b): h = Encoder(x),
where h = (h1, ..., hn) is the sequence of contex-
tualized representations for all input tokens. For
sequence labeling model, we use a feed-forward
network to compute the probability of whether that
token is belong to a potentially idiom expression:

\[ p_i = \text{softmax}(h_i) \] (1)

For span-based model, which are widely used in
reading comprehension task(Kwiatkowski et al.,
2019), where the answer is constrained to be a sin-
gle span from the input\(^1\). For that, two parameter-
ized functions (feed-forward networks), \( f_{\text{start}}(h_i) \)
and \( f_{\text{end}}(h_i) \), are used to compute a score for each
token, corresponding to whether that token is the
start or the end of the PIE. Last, the start probability

\[ y^s_{\text{start}} = \text{softmax}(f_{\text{start}}(h_i), ..., f_{\text{start}}(h_n)); \]

where \( p_{\text{start}} \in \mathbb{R}^{n \times 1} \), and \( y^s_{\text{end}} \) is the same. Train-
ing is done by minimizing cross entropy of the start
and end indices of the gold span, and at test time
the answer span is extracted by finding the indices
\((s, e)\):

\[ (s, e) = \arg\max_{s \leq e} p_{\text{start}}^s, p_{\text{end}}^e \] (4)

For the case that the sentences with no PIEs, we
add a special token as the index token, e.g., for
BERT we use '[CLS]' token, so the start and end
index both are 0.

2.2 Idiom-T5: Identification as Generation

We utilize the state-of-the-art text-to-text pre-
trained language model T5(Raffel et al., 2020),
which is pre-trained with a multitask objective
by prepending a task description before the in-
put text. We prepend the input sentence with a simple prompt: "extract idiom expression : ", and fine-tune the model with the source sen-
tence on the format “extract idiom expression : x
1, ..., w1, ..., w1; ..., w2, ..., w2”, where \( w_i \) is idiomatic
token. For example, for the input text "extract id-
iom expression : this dropped on me out of the
blue, i must admit this course.", the target sentence
is "output of the blue". For the case that the sen-
tences with no PIEs, we directly generate a text e.
We find the length of e could influence the model
performance as it involves a balance between the
numbers of the two classes (idiomatic and literal).
In our experiments, we set e as 'no idiom expres-
sion', which can achieve a better performance than
'none'. Finally, our training objective is to maxi-
mize the following likelihood:

\[ L = \sum_i \log P(x_i | x_{i-1}, ..., x_1; x) \] (4)

\( x \) is the source sentence, and \( x_i \in x \). For decoding,
we employ the greedy search, and set the beam size
as 5.

2.3 Data Augmentation using Sentence Shuffle

we observe that Idiom-T5 performs particularly
well in data scarce scenario, achieving 96% perfor-
ability only using 6% of full data. Hence, we aim
to utilize data augmentation to further improve its
performance. We propose a simple data augmentation method we called sentence shuffle: for the sentences contain several commas, we randomly shuffle the order of the sub-sentences separated by the comma; for the sentence with no comma, we randomly insert a comma anywhere outside the idiom expression(Karimi et al., 2021), and then apply the shuffle operation.

As the semantic integrity of idiomatic expression(Fraser, 1970), such shuffle hardly changes whether an expression is idiomatic or literal. However, the model needs to detect semantic compatibility when the sentences are not fluent enough, which can make the over-fitting of the model less serious and enhance its detection and generalization ability. Below is a pair example.

**Original**: He had been continually up the spout, or over the moon, about someone or something.

**Shuffled**: He had been continually up the spout, or over the moon, about someone or something, or over the moon.

### 3 Experiments

#### 3.1 Datasets

In order to test the models’ ability to identify unseen idioms, following(Zeng and Bhat, 2021), each dataset was split into train and test set in two ways: random(few-shot) and type-aware(zero-shot). In the random split, the sentences are randomly divided and the same PIE can appear in both sets, whereas in the type-aware split, the idioms in the test set and the train set do not overlap. In this work, we use the MAGPIE (Haagsma et al., 2020) dataset, which is the largest-to-date corpus of PIEs in English. MAGPIE consists of 1,756 PIEs across different syntactic patterns along with the sentences in which they occur (56,622 annotated data instances with an average of 32.24 instances per PIE). For random setting the number of training/val/test sets are train 3,533/4,457/4,451, and 35,531/4,461/4,449 for type-aware setting. The number of idiomatic and literal sentences are 34,129 and 10,321. Note that a expression might be idiomatic or literal in different sentences, which makes this task more challenging.

#### 3.2 Baseline Models

For the models described in section 2.1, we use BERT-Seq and BERT-Span to denote them respectively. Furthermore, we compare with the DISC(Zeng and Bhat, 2021), which is the state-of-the-art on the Magpie dataset, and the results of RNN-MHCA(Mao et al., 2019) and IlliniMET(Gong et al., 2020) reported in (Zeng and Bhat, 2021).

Our code is based on Huggingface’s transformers(Wolf et al., 2020). We evaluate the model on the development set and keep the best checkpoint for the final evaluation on test sets. We train our models using an AdamW optimizer(Loshchilov and Hutter, 2019), and for BERT based models, the learning rate is set as 3e-5, while 5e-5 is set for Idiom-T5. For the sentence shuffle augmentation, we generate an augmented sample for each sample in the dataset and then mix them up to form the final training dataset.

### 3.3 Evaluation Metrics

Following(Haagsma et al., 2020), we also use the two followings metrics to evaluate the performance of the models: 1) Classification F1 score (F1) measures the binary idiom detection performance at the sequence level with the presence of idioms being the positive class. 2) Sequence accuracy (SA) computes the idiom identification performance at the sentence level, where a sequence is considered as being classified correctly if and only if all its tokens are tagged correctly.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Model</th>
<th>0.5k(1.5%)</th>
<th>1k(3%)</th>
<th>2k(6%)</th>
<th>5k(15%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>I-T5base  +Aug</td>
<td>90.20</td>
<td>90.18</td>
<td>90.08</td>
<td>90.06</td>
</tr>
<tr>
<td></td>
<td>I-T5</td>
<td>90.18</td>
<td>90.08</td>
<td>90.06</td>
<td>91.90</td>
</tr>
<tr>
<td>SA</td>
<td>I-T5base  +Aug</td>
<td>67.60</td>
<td>73.13</td>
<td>75.25</td>
<td>77.23</td>
</tr>
<tr>
<td></td>
<td>I-T5</td>
<td>69.40</td>
<td>74.16</td>
<td>76.27</td>
<td>77.33</td>
</tr>
</tbody>
</table>

Table 1: The performance on different training size under type-aware setting using our proposed augmentation method compared with no augmentation. We ran the models with 3 different seed numbers and took the average score.
we can see that it’s not the increasing model parameters that leads to the better performance of Idiom-T5. Moreover, We experiment the other two generation language models UniLM(Dong et al., 2019) and BART(Lewis et al., 2020) to investigate whether the generation paradigm is always helpful. The results in Table 2 suggest that the generation paradigm is indeed helpful for some models, i.e., denoising pre-training models (BART and T5), and the excellent performance of Idiom-T5 may comes from its ’text-to-text format’ and supervised pre-train process. In others words, T5 is better at handling such unnatural generative task.

Furthermore, we find that compared with BERT-span, about 75% improvements in SA metric of Idiom-T5 comes from when the idiom length is 4 (e.g., in black and white) on the type-aware setting of MAGPIE dataset. This strongly demonstrates that Idiom-T5 has stronger ability to capture the semantic compatibility of unseen idioms that have long spans. We guess the reason is that for the generation paradigm, the encoding process focuses the encoder of Idiom-T5 capturing semantic information at the sentence level, and the decoding process constitutes a process of semantic contrast implicitly because the idiom itself needs to be input to the decoder of Idiom-T5. In the Appendix A we further discuss Idiom-T5 and its limitations.

## 5 Conclusion

In this work, we address the task of idiomatic expression identification by formulating it as a conditional text generation problem. Our Idiom-T5 shows remarkable improvements compared with the sequence labeling and span-based models. We further present a simple but effective data augmentation method to improve the performance of Idiom-T5 in data-scarce scenario. We hope our work can provide an alternative perspective to better address this tasks, or other tasks which also own the character of semantic non-compositionality.

---

### 4.2 Analyses

In this section, we analyze the effect of generation paradigm for idiomatic expression identification. As the parameters of models we list in Table 1, we can see that it’s not the increasing model parameters that leads to the better performance of Idiom-T5. Moreover, We experiment the other two generation language models UniLM(Dong et al., 2019) and BART(Lewis et al., 2020) to investigate whether the generation paradigm is always helpful. 

The results in Table 2 suggest that the generation paradigm is indeed helpful for some models, i.e., denoising pre-training models (BART and T5), and the excellent performance of Idiom-T5 may comes from its ’text-to-text format’ and supervised pre-train process. In others words, T5 is better at handling such unnatural generative task.

Furthermore, we find that compared with BERT-span, around 75% improvements in SA metric of Idiom-T5 comes from when the idiom length is 4 (e.g., in black and white) on the type-aware setting of MAGPIE dataset. This strongly demonstrates that Idiom-T5 has stronger ability to capture the semantic compatibility of unseen idioms that have long spans. We guess the reason is that for the generation paradigm, the encoding process focuses the encoder of Idiom-T5 capturing semantic information at the sentence level, and the decoding process constitutes a process of semantic contrast implicitly because the idiom itself needs to be input to the decoder of Idiom-T5. In the Appendix A we further discuss Idiom-T5 and its limitations.

### Table 2: Performance of models on the MAGPIE dataset evaluated by Classification F1 score (F1, %) and Sequence Accuracy (SA, %). We bold the best performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-span</td>
<td>91.79</td>
<td>70.66</td>
</tr>
<tr>
<td>UniLM</td>
<td>79.52</td>
<td>49.62</td>
</tr>
<tr>
<td>BART</td>
<td>89.30</td>
<td>73.34</td>
</tr>
<tr>
<td>T5</td>
<td>92.60</td>
<td>78.25</td>
</tr>
</tbody>
</table>

### Table 3: The performance on MAGPIE of three generative models under the type-aware setting of magpie dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN-MHCA†</td>
<td>-</td>
<td>95.51</td>
</tr>
<tr>
<td>IlliniMET†</td>
<td>-</td>
<td>86.54</td>
</tr>
<tr>
<td>DISC†</td>
<td>-</td>
<td>95.02</td>
</tr>
<tr>
<td>BERT-Seqbase</td>
<td>110M</td>
<td>94.70</td>
</tr>
<tr>
<td>BERT-Seqlarge</td>
<td>340M</td>
<td>94.88</td>
</tr>
<tr>
<td>BERT-Spanbase</td>
<td>110M</td>
<td>96.90</td>
</tr>
<tr>
<td>BERT-Spanlarge</td>
<td>340M</td>
<td>96.87</td>
</tr>
<tr>
<td>Idiom-T5base (Ours)</td>
<td>223M</td>
<td>96.50</td>
</tr>
<tr>
<td>Idiom-T5large (Ours)</td>
<td>739M</td>
<td>97.10</td>
</tr>
</tbody>
</table>
References


A The limitation of Idiom-T5

As discussed in (Zeng and Bhat, 2021), the MAGPIE dataset used in our experiments only label at most one PIE for each sentence even when there may be more than one, hence Idiom-T5 trained on MAGPIE can only generate one PIE at once. Therefore, there exists a few cases that model detects the alternative IE to the IE originally labeled as the ground truth. For example, the sentences "But an on-the-ball whisky shop could make a killing with its special ec-label malt scotch at £27.70 a bottle." contains two PIEs. Hence, the real performance for Idiom-T5 or DISC can be be a little higher.

However, because it is rare that a sentence contains multiple PIEs, obtaining such fully annotated data is difficult, but we have to deal with such cases. As a result, hits reflects the limitations of Idiom-T5. Theoretically, the sequence annotation model can handle such situation because of its classification at the token-level. However, from the actual effect, it is also difficult for DISC (Zeng and Bhat, 2021) to accurately detect multiple PIEs due to the lack of effective supervision signals. We argue that we can randomly combine multiple sentences containing PIEs in the training set into a longer sentence, and then the goal of Idiom-T5 is to detect these multiple PIEs. For example, as we described in Section 2.2, the target output of Idiom-T5 is "on-the-ball ; make a killing" in the above example, rather than "make a killing". Through this automatic construction method, we hope Idiom-T5 can generalize its detection ability to the test set, as there may be contextual relationships between multiple PIEs in the test set, but not during training time. We leave this as our future work.