# NLP@UIT at FigLang-EMNLP 2022: A Divide-and-Conquer System For Shared Task On Understanding Figurative Language

**Khoa Thi-Kim Phan, Duc-Vu Nguyen, Ngan Luu-Thuy Nguyen**
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University Ho Chi Minh City, Vietnam
{khoaptk, vund, ngannlt}@uit.edu.vn

## Abstract

This paper describes our submissions to the EMNLP 2022 shared task on Understanding Figurative Language as part of the Figurative Language Workshop (FigLang 2022). Our systems based on pre-trained language model **T5** are divide-and-conquer models which can address both two requirements of the task: 1) classification, and 2) generation. In this paper, we introduce different approaches in which each approach we employ a processing strategy on input model. We also emphasize the influence of the types of figurative language on our systems.

## 1 Introduction

Recent years have witnessed the great rise of Artificial Intelligence (AI). Due to the performance of AI, many downstream tasks from any fields are solved efficiently. One of the central topic in AI is Natural Language Understanding (NLU) in which Natural Language Inference (NLI) or Recognizing Textual Entailment (RTE) plays an important role, which was pointed out in (MacCartney and Manning, 2008).

While RTE was defined as a task of determining whether a natural language hypothesis *h* can be inferred from a given premise *p* (MacCartney, 2009), Figurative Language Understanding (FLU) was considered as a task of determining whether any figure of speech depends on a non-literal meaning of some or all of the words used (Chakrabarty et al., 2022). Therefore, FLU can be framed as a kind of RTE task (Chakrabarty et al., 2022; Stowe et al., 2022).

In addition, the EMNLP 2022 shared task requires not only to generate the label (entail/contradict), but also to generate a plausible explanation for the prediction, whose example is shown in Table 1. Especially, the entail/contradict label and the exploration are related to the meaning of the figurative language expression. This is a challenging task that require to propose a approach that could tackle both tasks: 1) classification, 2) generation.

| | |
|---|---|
| Premise | The place looked impenetrable and inescapable |
| Hypothesis | The place looked like a fortress. |
| Label | Entailment |
| Explanation | A fortress is a military stronghold,<br>hence it would be very hard to walk into,<br>or in other words impenetrable and inescapable. |

Table 1: Examples of relations between a premise and a hypothesis: E (Entailment), C (Contradiction).

Over the past few years, a number of high-performance systems have been created solving several NLP tasks based on pre-trained transformer models (Vaswani et al., 2017; Devlin et al., 2019; Lewis et al., 2019; Raffel et al., 2020b). However, there have still been very few works related to figurative language due to the lack of high-quality datasets and the challenge of this task.

Therefore, thanks to the exclusive dataset of the shared task, in this paper, we advocate different approaches which are mainly based on pre-trained language model **T5** (Raffel et al., 2020b), combining to employ various input processing strategies to tackle the task.

In this paper, we conduct an investigation into the benefit of using state-of-the-art seq2seq pre-trained language models (T5) to evaluate figurative language understanding task in EMNLP 2022. We also employ a divide-and-conquer model with different potential input processing strategies to improve the performance of our system. Then, we point out the importance of the types of figurative language in this task.

## 2 System Description

In all our submissions, we considered both two tasks: the NLI task, and the explanation generation task as two seq2seq tasks. Therefore, we fine-tuned

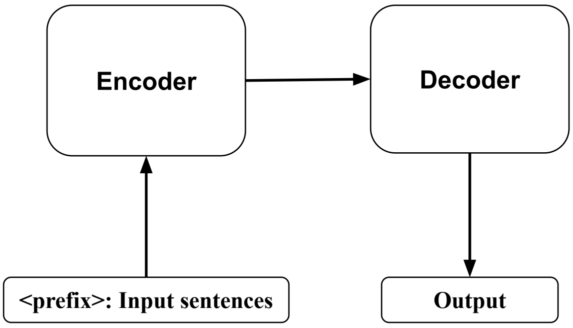
*Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 150 - 153
December 8, 2022 ©2022 Association for Computational Linguistics

two tasks jointly as a simultaneous computation model which first predicts label, and then the explanation. In addition, we also used the attribute about types of Figurative Language across the data as a predictor and treated it as seq2seq tasks. Therefore we have 3 component models based on fine-tuning pre-trained model T5 (Raffel et al., 2020b): NLI predictor, Type predictor, and Generator.

### 2.1 T5

T5 transformer is a encoder-decoder model or sequence-to-sequence model. It is a “unified framework that converts every language problem into a text-to-text format” (Raffel et al., 2020b). Compared to other transformers which take in natural language data by converting to corresponding numerical embeddings, T5 takes in data in the form of text, and also produce the text as an output. This text-to-text nature does not require any the change of hyper-parameters and loss functions when learn NLP tasks (Grover et al., 2020). Furthermore, T5 has been trained on a multi-task mixture of unsupervised and supervised tasks in which include our NLI task and generation task. Therefore, T5 model is one of the most prominent pre-trained models that we can use.

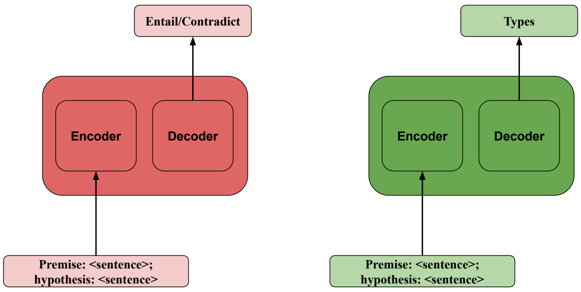
Figure 1: Overview of input and output of T5.



### 2.2 NLI predictor and Type predictor

In this two component models, the premise and hypothesis sentences are concatenated and fed to the encoder, then while the decoder of NLI task is the label prediction (entail/contradict), the decode of Type predictor is the type prediction (Paraphrase, Sarcasm, Simile, Metaphor, Idiom). The overview of two component models are shown in Fig.2

Figure 2: Overview of two component models. Red diagram is NLI predictor, the green diagram is Type predictor.


Entail/Contradict
Encoder
Decoder
Premise: <sentence>; hypothesis: <sentence>
Types
Encoder
Decoder
Premise: <sentence>; hypothesis: <sentence>

### 2.3 Generator

We employed different input processing strategies each submission in the Generator. Specifically, in the first submission, we simply used the premise and hypothesis sentences as a input of the encoder as same as NLI predictor did. However, the performance of the model is not too well, so we tried to add valuable attributes such as NLI predictor, and Type predictor to the left of the input of the encoder. Therefore, we conducted experiments for submission 2, 3, 4 by adding a NLI predictor, a Type predictor, and a NLI predictor + a Type predictor to the left of the input, respectively. Besides, The 5th submission is similar to the 3rd submission, except the parameters of the model. The model is depicted in Fig.3.

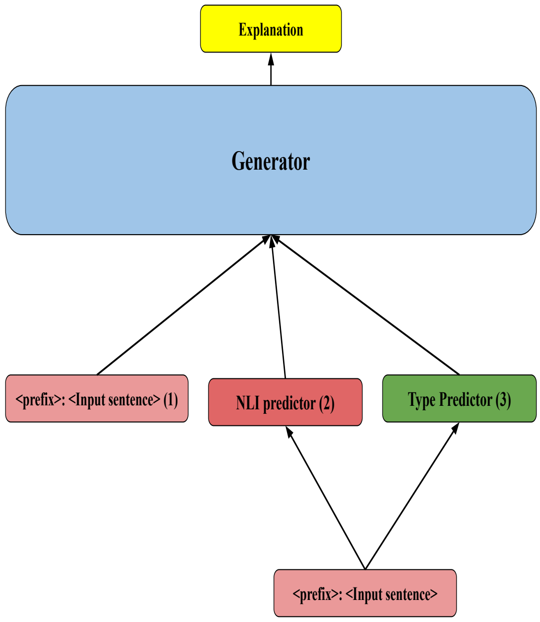
## 3 Experiments

### 3.1 Experimental setting

Following the given evaluation metrics, in all our experiments, we report the Accuracy@60 based on evaluation scripts from the task organizing committee.

As described in **Section 2**, our approaches depends on pre-trained language model T5. We use a model namely T5$_{large}$ downloaded from the Hugging Face library (Raffel et al., 2020a). The network's parameters are optimized using the AdamW (Loshchilov and Hutter, 2017) and a linear learning rate scheduler, which are suggested by the Hugging Face default setup. The hyperparameters that we tune include the number of epochs, batch size, and learning rate. In particular, we set batch size of 32, and learning rate of 3e-4 for all component models. For NLI predictor and Type predictor, we use 20 epochs. For Generator, the model is trained

Figure 3: Overview of submissions. While the component (1), (1)+(2), (1)+(3) is considered as a input model of the 1st, 2nd, 3rd submission respectively, the 4th one has all 3 components as a input model.



| Submissions | Input model | Score | Size (bytes) |
|---|---|---|---|
| 1 | premise, hypothesis | 58.26 | 39154 |
| 2 | NLI predictor, premise, hypothesis | 57.93 | 38015 |
| 3 | Type predictor, premise, hypothesis | 60.53 | 42532 |
| 4 | NLI predictor, Type predictor,<br>premise, hypothesis | 59.80 | 37763 |

Table 2: Official results of our system on test dataset.

on 40 epochs. All experiments in this paper are conducted on Google Colab Pro.

### 3.2 Result and Discussion

For producing the results on the test dataset, we splited the training dataset into the training dataset and development dataset with 7300 samples, and 200 samples respectively for fine-tuning the pre-trained language model T5$_{large}$.

Our latest system achieved the official score 60.53 which ranked 3rd on the shared task. On each of the submissions, the systems obtained scores 58.26, 57.93, 60.53, and 59.80 respectively. Table 2 gives the detailed results of each submissions.

Comparing the detailed scores, we found that our submitted systems varied in performance mainly due to the difference of input model of the submissions. As described in Table 2, the system in which the input of model is the combination of NLI predictor, premise and hypothesis performed the worst, while the one which has the type predictor combining with input sentences (premise and hypothesis) outperformed the rest of our experiments. Therefore, the types of figurative language were indicated to be an integral role in understanding figurative language.

Depending on the input models, the Generator has different outputs, as shown in the Table 3. Compared to the models which add only one component into the input models: NLI predictor or Type predictor, the model of submission 4 had more information that consists of two input sentences, NLI predictor, and Type Predictor. However, the Generator did not produce adequate explanations as we expected. Therefore, the strategy including more information may not be a good choice when generating outputs in this case. Despite that, more efforts are required to explore the real reason behind the results, then we can learn and employ the input processing strategies reasonably to improve the performance of the system.

## 4 Conclusion and Future Work

In this paper, we have presented our system for the EMNLP 2022 Shared Task on the Figurative Language Understanding. Our systems are built on fine-tuning pre-trained language model T5 with different input processing strategies, which is a divide-and-conquer model which integrated two or three components: NLI predictor, Type predictor,

| Sample | Explanation | NLI predictor | Type Predictor |
|---|---|---|---|
| "premise": "I stubbed my toe last night and cursed angrily."<br>"hypothesis": "Stubbing my toe last night and cussing out loud made me so happy."<br>"Predicted Label": Contradiction | "Stubbing one's toe is usually a very painful experience and can result in people feeling angry and cursing loudly which is not a happy feeling." | | |
| | "Stubbing your toe and cursing loudly is not a good thing because it can cause pain and discomfort." | X | |
| | "Stubbing one's toe and cursing loudly is not a good thing and so being happy about it cannot be justified." | | X |
| | "Stubbing your toe and cursing loudly is not a good way to spend a night in bed and so someone who is happy about it cannot be considered rational." | X | X |

Table 3: Examples of explanation produced by the systems.

and Generator. The performance of models are relied how successful the Type predictor is, which means the attribute about types of Figurative Language should be considered as an integral factor of the input of model.

Due to limited time and resources, we had not conducted thorough enough experiments to get better results, but the system and the involvement in this challenge bring us a good groundwork for further study. In the future, we plan to expand the experiment by employing and fine-tuning other pretrained language models. Furthermore, we may also explore different strategies making the most of what we have for the input of models.

## References

Tuhin Chakrabarty, A. Saakyan, Debanjan Ghosh, and Smaranda Muresan. 2022. Flute: Figurative language understanding through textual explanations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Khushnuma Grover, Katinder Kaur, Kartikey Tiwari, Parteek Kumar, et al. 2020. Deep learning based question generation using t5 transformer. In *International Advanced Computing Conference*, pages 243–255. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*.

Bill MacCartney. 2009. *Natural language inference*. Stanford University.

Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Kevin Stowe, Prasetya Utama, and Iryna Gurevych. 2022. Impli: Investigating nli models' performance on figurative language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5375–5388.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.