HIT at SemEval-2022 Task 2: Pre-trained Language Model for Idioms Detection

Zheng Chu[†‡], **Ziqing Yang**[‡], **Yiming Cui**[†‡], **Zhigang Chen**[‡], **Ming Liu**[†]
[†]Research Center for SCIR, Harbin Institute of Technology, Harbin, China
[‡]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China
[†]{zchu,ymcui,mliu}@ir.hit.edu.cn
[‡]{zhengchu,zqyang5,ymcui,zgchen}@iflytek.com

Abstract

The same multi-word expressions may have different meanings in different sentences. They can be mainly divided into two categories, which are literal meaning and idiomatic meaning. Non-contextual-based methods perform poorly on this problem, and we need contextual embedding to understand the idiomatic meaning of multi-word expressions correctly. We use a pre-trained language model, which can provide a context-aware sentence embedding, to detect whether multi-word expression in the sentence is idiomatic usage.

1 Introduction

The goal of the SemEval-2022 Task2 (Tayyar Madabushi et al., 2022) SubtaskA is to detect whether a multi-word expression in a sentence is idiomatic in usage. It is a multilingual task and consists of three languages: English, Portuguese and Galician.

Multi-word expressions (MWEs) are expressions that consist of at least two words and are syntactically or semantically specific. The semantics of MWEs are usually divided into two types, (i) the combination of literal meanings of each word in the phrase or (ii) inherent usages (e.g., idiomatic meaning). Understanding the semantic meaning of a sentence requires the correct identification of the MWE in the sentence. Table 1 contains one case for each of the two usages.

Traditional non-contextual word embedding models, such as word2vec (Mikolov et al., 2013), perform poorly at this task. Simple superposition of non-contextually word embeddings does not correctly express the semantics of idiomatic phrases. Therefore, contextual embedding models (Conneau et al., 2020; Devlin et al., 2019) are required to correctly understand the meaning of multi-word expressions in idiomatic usage.

We used large-scale cross-lingual pre-trained language models, multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), with a softmax classifier on top of the pre-trained LM to train a binary classification model. The training data are processed before training, and regularization dropout (Liang et al., 2021), adversarial training (Miyato et al., 2017; Madry et al., 2018) are used in the training process. In addition, we observed the training data and found an interesting phenomenon that we can get better results by post-processing after the training using heuristic rule.

Literal	When removing a **big fish** from a net, it should be held in a manner that suports the girth.
Idiomatic	It was still a respectable finish for both Fadol and Nayre, who were ranked outside the top 500 in the world but caught some **big fish** along the way

Table 1: Examples of idiomatic and non-idiomatic usage

2 Background

2.1 Task Description

Task 2 contains two subtasks, SubtaskA is idiom detection, and SubtaskB is similarity scoring of texts containing idioms. This article focus on SubtaskA. SubtaskA contains two settings, zero-shot and one-shot.

- Zero-shot: Multi-word expressions that appear in test data do not appear in training data.
- One-shot: Every multi-word expressions that appeared in the test data appeared in training data at least once.
- Data Restriction: Under zero-shot setting, we can only use zero-shot training data, and we can use both zero-shot and one-shot training data under one-shot setting. Test data is same for both settings.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 221 - 227

2.2 Data Details

In this section, we will describe the characteristics of the training data and test data. The official data includes eight columns, which are DataID, Language, MWE, Setting, Previous, Target, Next, Label.

2.2.1 Language

The test data includes 916, 713, 713 entries in English, Portuguese and Galician, respectively.

There are only English and Portuguese examples in the training data of zero-shot setting, which means that in the testing phase, the model requires zero-shot transfer of Galician with the learned knowledge on other two languages. Any MWE in zero-shot training data will not appear in the test data.

In one-shot training data, there are 73 entries in each of the three languages, which is relatively small compared to zero-shot training data. Any MWE in the one-shot training data will appear in the test data.

2.2.2 Data Length

We counted the average length of the data to facilitate appropriate truncation when using a pre-trained model.

The average, median, max length of target sentences after tokenizer corresponding to the pre-train model are 42.6, 195, 40, respectively. Over 90% of sentences are 64 or less in length.

The average, median, max position that MWE occurs in sentences after tokenizer is 18.9, 89, 15, respectively, and over 90% of sentences are 37 or less in position.

2.3 Related Work

So far, there has been extensive research about idioms detection. (Zeng and Bhat, 2021) propose a multi-stage neural architecture with attention flow. (Garcia et al., 2021a,b) probs idiomaticity in vector space and propose NCTTI dataset. (Do Dinh et al., 2018) propose a multi-task learning method. (Tayyar Madabushi et al., 2021) present a multilingual idiom detection dataset, which will be used as this SemEval-2022 idioms detection track.

Pre-trained word embedding can capture syntactic and semantic information from large amounts of unlabeled data, which has been a standard part of natural language processing task (Mikolov et al., 2013; Pennington et al., 2014). However, each of these methods can only obtain a fixed, non-contextual vector representation for each word which makes it difficult to convey the correct meaning of polysemous words. Due to the disadvantages of non-contextual embedding, recent work has begun to focus on contextual embedding, typical cases are context2vec (Melamud et al., 2016), ELMo (Peters et al., 2018), BERT (Devlin et al., 2019).

The most commonly used in contextual embedding is the pre-trained language model (Devlin et al., 2019; Conneau et al., 2020). These models perform self-supervised training through mask language modeling, next sentence predicting, and other objectives in hundreds of millions of unlabeled datas. Benefiting from multilingual training data, these models have cross-language capabilities. Pre-trained models are gradually taking the place of pre-trained word embeddings as the new paradigm for natural language processing.

3 System Overview

Figure 1 depicts the flow chart of the whole system. We first preprocess the data, tokenizing and then feed into a pre-trained model to get hidden states. Apply some pooling method to get fixed length sentence representation to train a softmax binary classifier. After that, the prediction results of the model are post-processed. In the training process, contrastive learning, adversarial training, regularized dropout, etc. are used. Table 2 is then used as an example to introduce the data preprocessing process of the system.

3.1 Baseline

The baseline method below refers to the method in paper (Tayyar Madabushi et al., 2021).

- In the zero-shot setting, Multilingual BERT is trained on zero-shot data, using the context without idiom as an additional feature.
- In the one-shot setting, Multilingual BERT is trained on combination of the zero-shot and one-shot data, excluding the context and adding the idiom as an additional feature.

3.2 Data Preprocessing

3.2.1 Truncation

The data provides the context of the target sentence. We only use target sentences for training because we found that if we concatenate previous, target,

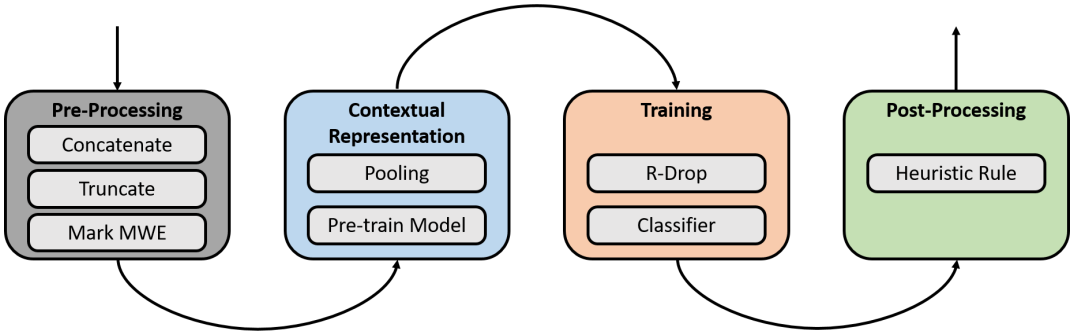

Figure 1: System flow diagram

and next together, the sentence length will be too long, which slows down training and harms performance. We guess that to distinguish whether the MWE is an idiomatic usage, we only need to focus on words near the MWE, too long sentences will introduce unnecessary distractions.

According to the length statistics in the previous chapter, 128 is used as the maximum truncation length of the pre-trained model, which can ensure that most sentences will not be truncated and keep the sentence length as small as possible.

3.2.2 MWE Marking

Following the baseline method, we use the tokenizer's [SEP] token to mark the MWE in the sentence. Unlike the baseline method, we only mark MWEs without deformation. Proper nouns are usually non-idiomatic usage, and are often deformed. Pre-trained models can recognize proper nouns well, so we do not mark the deformed MWEs. The results also show that this gives better performance.

Example in Table 2, MWE 'milk tooth' in sentence "Her latest pamphlet **Milk Tooth**, published by Rough Trade Books, is a collection of thwarted escape plans for a too-heavy world" is capitalized, according to our rules, the MWE in this sentence is not marked. If the MWE in the sentence is not deformed, the [SEP] token will be used to mark the MWE, just like [SEP]**milk tooth**[SEP].

3.3 Model

3.3.1 Pre-trained Model

We tried different multilingual pre-trained models, including mBERT and XLM-RoBERTa, and XLM-RoBERTa consistently outperforms mBERT. In addition, we also try to use different size models, including mBERT-base-cased, XLM-RoBERTa-base, XLM-RoBERTa-large, and bigger models lead to better performance.

3.3.2 Classifier

Different pooling methods are used for hidden states of different layers, including mean pooling, max pooling, [CLS], and token-level pooling. The results show that different pooling methods have little effect on the results. For simplicity, [CLS] is used as the sentence representation.

For token-level pooling, we pool the vectors of MWE positions in hidden states to obtain the final sentence representation, which harms the performance.

After pooling on the pre-trained model, fixed-length sentence representation is obtained. This is followed by a full connection layer with dropout (Srivastava et al., 2014) and a softmax classifier.

3.3.3 Regularized Dropout

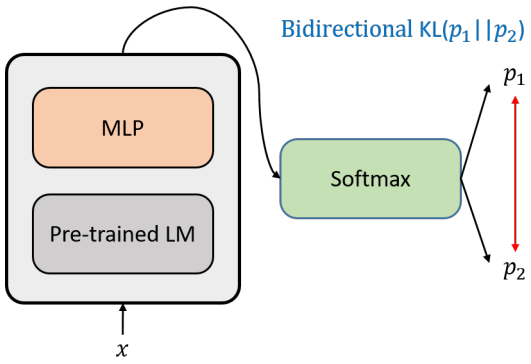

Figure 2: Regularized Dropout

Deep neural networks usually use dropout (Srivastava et al., 2014), but the use of dropout introduces inconsistency between train and inference. R-drop (Liang et al., 2021) is a means of regularization in the training process to mitigate this inconsistency.

A sample output through the pre-trained model, MLP and softmax can be regarded as a probability distribution. R-drop performs two independent

MWE	milk tooth
Previous	A ritual sacrifice from the 19th century is vividly relieved.
Target	Her latest pamphlet **Milk Tooth**, published by Rough Trade Books, is a collection of thwarted escape plans for a too-heavy world.
Next	In these poems of trauma and transformation, the present throbs with unfinished histories.
Label	1

Table 2: Training data example (useless columns have been removed)

forward calculations for each sample, obtaining two outputs probability distribution. Due to the dropout, these two outputs will be slightly different, introducing inconsistency. To mitigate this, the bi-directional KL-divergence is calculated as a penalty between these two outputs probability distributions. In the following equation, y_i represents the label, x_i represents the input, and the superscript represents two independent forward operations. p_i^1, p_i^2 represent probability distribution obtained from two independent forwards.

$$\mathcal{L}_{CE}^{i} = CE(x_i^1, y_i) \; + \; CE(x_i^2, y_i) \tag{1}$$

$$\mathcal{L}_{KL}^{i} = \frac{1}{2}(\mathcal{D}_{KL}(p_i^1 || p_i^2) + \mathcal{D}_{KL}(p_i^2 || p_i^1)) \tag{2}$$

$$\mathcal{L}^{i} = \mathcal{L}_{CE}^{i} + \alpha \cdot \mathcal{L}_{KL}^{i} \tag{3}$$

3.3.4 Post Processing

We found an interesting phenomenon in the training data. Some MWEs in one-shot training data have only one category label, and most of these MWEs corresponding to entries in the dev data have the same label as in the training data. Some proper nouns are labeled with idiomatic meanings, however some with literal meanings. These labeling inconsistencies may cause problems in the learning of the model, so we design a heuristic rule. On top of the model prediction results, if there is only one label for a certain MWE in the training set, then replace all the predictions for that MWE in the test set with whichever label appears in the training set.

4 Experiment

4.1 Hyperparameters

We use the Huggingface Transformers (Wolf et al., 2019) implementation of mBERT and XLM-RoBERTa. During the training, the learning rate schedule strategy is warmup of first 10% steps with cosine learning rate decay in rest steps. For zero-shot and one-shot, we use learning rates of 1e-5 and 3e-5, respectively, and one-shot is continued training on the well-performing zero-shot model. The mini-batch size is 32. The coefficient of R-drop is chosen from 0, 1, 2, 4. We train a total of 20 epochs and save the best-performing checkpoints on the development set. All models are trained on one single NVIDIA Tesla V100 GPU.

Model	Zero-shot	One-shot
mBERT$_{base}$ w/ C w/ I	74.90	84.78
mBERT$_{base}$ w/ C w/o I	70.59	76.98
mBERT$_{base}$ w/o C w/ I	**75.31**	**85.76**
mBERT$_{base}$ w/o C w/o I	70.76	82.59

Table 3: Context and idiom effects on the development set results. C: context. I: Idiom. (Macro F1 $\times$ 100)

Model	Max Len	Zero-shot	One-shot
mBERT$_{base}$	128	**76.31**	**87.97**
mBERT$_{base}$	192	75.62	87.96
mBERT$_{base}$	256	75.64	86.24

Table 4: Effect of different maximum sentence lengths on the development set results. (Macro F1 $\times$ 100)

4.2 Evaluation Metrics

SubtaskA is evaluated using the Macro F1 score between the gold labels and model predictions.

$$Macro\ F1 = 2 \cdot \frac{precision \times recall}{precision + recall} \tag{4}$$

4.3 Hyperparameters Selection

We compare different pre-trained models, max length, and whether to use MWE and contexts.

4.3.1 Context and Idiom

We compare the performance of the model with and without context in Table 3, the effect of marking idioms on the results. Moreover we conduct experiments on BERT$_{base}$, using [CLS] as pooling, with the max sentence length set to 128. The method of marking MWEs here follows the baseline.

Experiments show that ignoring context and mark idioms gives better results, and this setting will be continued for future experiments.

Model	Zero-shot	One-shot
mBERT$_{base}$	75.31	87.97
XLM-R$_{base}$	76.99	89.15
XLM-R$_{large}$	**78.17**	**91.84**

Table 5: Effect of different pre-trained models on the development set. (Macro F1 $\times$ 100)

4.3.2 Max Sentence Length

In this section, a comparison is made between the cases with different maximum sentence lengths. The model follows the previous setting, with idioms marked and context ignored, using first-last-avg as pooling for training.

Our original hypothesis is that performance and speed are a trade-off as the maximum sentence length increases. In contrast, Table 4 shows that a maximum sentence length of 128 is sufficient in terms of speed and performance. We do not test a smaller maximum sentence length because further reduction might cause parts of the sentence to be truncated, harming the performance.

4.3.3 Pre-trained Model

We compared mBERT$_{base}$, XLM-RoBERTa$_{base}$, and XLM-RoBERTa$_{large}$, and results in Table 5 demonstrated that XLM-RoBERTa outperformed mBERT, and the large model performed better than base model. In addition, our final submission results were obtained using XLM-R$_{large}$.

From the results, we found that the performance improvement is evident as the model size grows, and there is no bottleneck yet. Therefore, increasing the model size may be a simple and effective way.

5 Results

5.1 System Performance

Our final results use model fusion on thirteen models. The zero-shot model finished fourth with an F1 of 77.15, and the one-shot model finished first with an F1 of 93.85.

5.2 Ablation Study

Table 6 provides the results of the ablation study. The baseline of the ablation experiment follows the hyperparameters of the experiment chapter, except that the model is replaced with XLM-R$_{large}$.

Model	Zero-shot	One-shot
XLM-R$_{large}$	78.05	91.02
+mark MWE	78.17	91.84
+contrastive pre-train	-	89.95
+contrastive auxiliary	76.30	88.05
+AEDA	79.09	89.76
+AT	79.78	92.47
+R-drop	**80.34**	**92.91**
+post-processing	-	**93.73**

Table 6: Ablation experiments on the development set. (Macro F1 $\times$ 100)

5.2.1 Mark MWE

First, we change the method of marking MWE in the baseline. We use [SEP] token for tagging only if the MWE in the sentence is the same as the MWE provided in the data. The reason for this is that the organizer's rule is to label proper nouns as literal meaning, and proper nouns are usually deformed with initial capitalization. The pre-trained model can distinguish proper nouns well under the training of a large amount of corpus.

5.2.2 Adversarial Training

Adversarial training (Miyato et al., 2017; Madry et al., 2018) is a way to enhance the robustness of neural networks by adding small perturbations to the samples to interfere with the predictions of the model. In our experiments, the results in Table 6 show that adversarial training significantly improves performance.

5.2.3 R-drop

R-drop is a regularization tool that aims to maintain the consistency of model prediction and training while using dropout by adding bi-directional KL-divergence as a penalty term. After adding R-drop, the performance is significantly improved, exceeding the adversarial training. In Table 6, under zero-shot setting, the relative improvement is 2.17 and 0.56 compared to baseline and adversarial training, respectively, and this improvement is 1.07 and 0.44 under one-shot setting, respectively.

5.2.4 Heuristic Rule

We used the heuristic rule mentioned in 3.3.4 for replacement under the one-shot setting, and as shown in the Table 6, the relative improvement is 0.82 percentage points.

5.2.5 Negative Results

Contrastive Learning: Recently, contrastive learning has been a hot topic in NLP, especially in sentence representation learning. Inspired by SimCSE (Gao et al., 2021), we use contrastive learning before and during training classification, using the data from subtaskA for contrastive pre-train and apply contrastive loss during training as an auxiliary training objectives, respectively. Unfortunately, as shown in Table 6, both of these methods make the results much worse.

Data Augmentation: Due to lack of training data, we use data augmentation for expansion. We used AEDA (Karimi et al., 2021) as a means of data augmentation, which is a straightforward data augmentation, by adding punctuation marks to the sentences. The results showed that a particular improvement was achieved under zero-shot setting, but a decrease was achieved under one-shot setting.

6 Conclusion

We continuously improve the model performance by improving the MWE marking method, using larger pre-trained models, adding regularization terms and heuristic rules. Inspired by pre-trained models, we can find that future work needs to focus more on external data besides training data, especially data with specific idioms, such as idiom dictionaries, because the size of the external data is much larger than the training data, which has not been fully exploited.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Erik-Lân Do Dinh, Steffen Eger, and Iryna Gurevych. 2018. Killing four birds with two stones: Multi-task learning for non-literal language detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1558–1569, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021a. Assessing the representations of idiomaticity in vector models with a noun compound dataset labeled at type and token levels. 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)*, pages 2730–2741, Online. Association for Computational Linguistics.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021b. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. *CoRR*, abs/2106.14448.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference*

on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Harish Tayyar Madabushi, Edward Gow-Smith, Marcos Garcia, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2022. SemEval-2022 Task 2: Multilingual Idiomaticity Detection and Sentence Embedding. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Ziheng Zeng and Suma Bhat. 2021. Idiomatic Expression Identification using Semantic Compatibility. *Transactions of the Association for Computational Linguistics*, 9:1546–1562.