Hitachi at SemEval-2022 Task 2: On the Effectiveness of Span-based Classification Approaches for Multilingual Idiomaticity Detection

Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki and **Yasuhiro Sogawa**
Research and Development Group, Hitachi, Ltd.
Kokubunji, Tokyo, Japan
{atsuki.yamaguchi.xn, gaku.morio.vn,
hiroaki.ozaki.yu, yasuhiro.sogawa.tp}@hitachi.com

Abstract

In this paper, we describe our system for SemEval-2022 Task 2: *Multilingual Idiomaticity Detection and Sentence Embedding*. The task aims at detecting idiomaticity in an input sequence (Subtask A) and modeling representation of sentences that contain potential idiomatic multiword expressions (MWEs) (Subtask B) in three languages. We focus on the zero-shot setting of Subtask A and propose two span-based idiomaticity classification methods: MWE span-based classification and *idiomatic* MWE span prediction-based classification. We use several cross-lingual pre-trained language models (InfoXLM, XLM-R, and others) as our backbone network. Our best-performing system, fine-tuned with the span-based idiomaticity classification, ranked fifth in the zero-shot setting of Subtask A and exhibited a macro F_1 score of 0.7466.

1 Introduction

SemEval-2022 Task 2 (Tayyar Madabushi et al., 2022) involves detecting idiomaticity in a given sentence (Subtask A) and learning effective representations of sentences that may contain idiomatic multiword expressions (MWEs) (Subtask B) in three languages: English, Portuguese, and Galician. Processing idiomaticity in a sequence correctly is an essential task in natural language processing (NLP), as idiomatic expressions are a key component of natural languages. Its high performance will contribute to various downstream tasks, such as sentiment analysis, information retrieval, and machine translation (Hashempour and Villavicencio, 2020; Tayyar Madabushi et al., 2021).

In this work, we propose two different approaches for multilingual idiomaticity detection (Subtask A) that take advantage of MWE span-based features. We use several cross-lingual pre-trained language models (InfoXLM (Chi et al., 2021a), XLM-R (Conneau et al., 2020), and others) and exploit their MWE span representations for classification, instead of using a special classification token (`[CLS]`), which typically corresponds to the first input token. Our concept is that these models should be able to focus more on the idiomaticity of an MWE in a given sequence by using its span representation rather than using the `[CLS]` token for classification, potentially resulting in a better detection performance.

Our main findings in the shared task are in threefold.

1. The span-based idiomaticity classification method is highly effective compared to the standard `[CLS]`-based sequence classification approach adopted in various BERT (Devlin et al., 2019)-like models (Liu et al., 2019; Lan et al., 2020; Clark et al., 2020).

2. Detecting idiomaticity in Galician with no training data available is challenging even with state-of-the-art cross-lingual pre-trained language models.

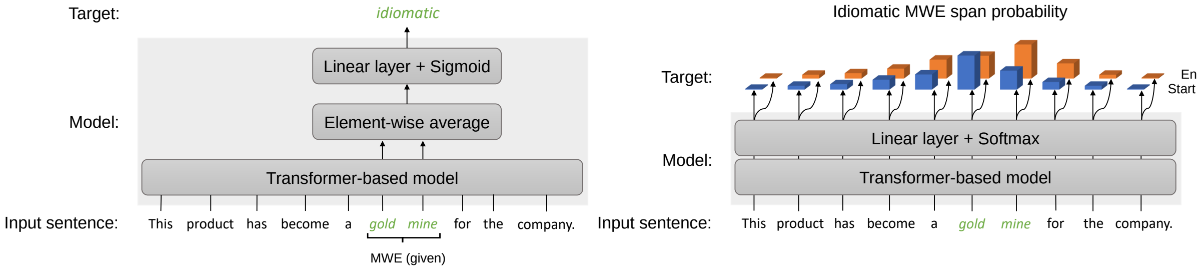
3. Utilizing adjacent contexts with a target sentence is not always the best option for idiomaticity detection, even though it improves the baseline performance.

Consequently, our best-performing system, using the span-based classification, ranked fifth among 20 systems in the zero-shot setting of Subtask A and showed a macro F_1 score of 0.7466 on the test set.

2 Background

Idiomaticity Detection While the task of idiomaticity detection with respect to MWEs is not new, it is still considered challenging because state-of-the-art language representation models heavily depend on the principle of compositionality (Pelletier, 1994), which idioms do not follow, due to their tokenization methods (Kudo, 2018; Sennrich

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 135 - 144
July 14-15, 2022 ©2022 Association for Computational Linguistics



(a) Span-based idiomaticity classification

(b) Span prediction-based idiomaticity classification

Figure 1: Overview of two proposed approaches for detecting idiomaticity.

et al., 2016). To overcome this problem, some studies (Hashempour and Villavicencio, 2020; Garcia et al., 2021) have regarded an MWE as a single token motivated by the assumption that people recognize an idiom as a single token (Sinclair et al., 1991). Alternatively, others have tried to utilize the adjacent contexts of MWEs as inputs and have demonstrated the effectiveness of this against tasks targeting verb-noun constructions (Sporleder and Li, 2009; Salton et al., 2016; King and Cook, 2018) and noun compounds (Tayyar Madabushi et al., 2021). This paper also utilizes adjacent contexts for classification but proposes new idiomaticity detection approaches in which the span information of an MWE plays an important role.

Cross-lingual Pre-trained Language Models Cross-lingual pre-trained language models have shown promising results in multilingual NLP tasks since the emergence of multilingual BERT (mBERT) (Devlin et al., 2019). In general, they are pre-trained with either multilingual masked language modeling (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020; Chung et al., 2021) or translation language modeling (Conneau and Lample, 2019). The difference between the two is that the former uses monolingual sentences while the latter utilizes concatenated parallel sentences for inputs. The state-of-the-art InfoXLM (Chi et al., 2021a) further utilized contrastive learning, where a model needs to distinguish a correct translated sample from negative ones. Our approach uses several cross-lingual pre-trained language models, including InfoXLM, XLM-R, XLM-Align (Chi et al., 2021b), and RemBERT (Chung et al., 2021), to utilize multilingual idiomaticity data efficiently.

3 Task Description

We briefly describe a multilingual idiomaticity detection task (Subtask A). Given a sentence composed of n words $\mathcal{S}_{\text{target}} = [w_1, \ldots, w_n]$ that contains an m-word MWE $\mathcal{W} = [w_1^{\text{MWE}}, \ldots, w_m^{\text{MWE}}]$, $\mathcal{S}$'s preceding sentence $\mathcal{S}_{\text{prev}}$, and succeeding sentence $\mathcal{S}_{\text{next}}$, the task is to classify if $\mathcal{W}$ is *idiomatic* (0) or not (1). The task dataset is based on Tayyar Madabushi et al. (2021) and contains *Galician* in addition to English and Portuguese. Each sample consists of $\mathcal{S} = [\mathcal{S}_{\text{prev}}; \mathcal{S}_{\text{target}}; \mathcal{S}_{\text{next}}]$, $\mathcal{W}$, a language type $lang \in \{\text{“EN”}, \text{“PT”}, \text{“GL”}\}$, and an idiomaticity label $y_{\text{MWE}} \in \{0, 1\}$. In the zero-shot setting, participants do not have any training samples for Galician and are only allowed to use the officially provided training and development sets for training. They must also use the same approach for all samples except language and can submit up to five systems for evaluation.

4 System Overview

Our system relies on cross-lingual pre-trained language models (InfoXLM and XLM-R and others) and classifies samples using either span-based classification or span prediction-based classification. We fine-tune several pre-trained language models and obtain final predictions by using an ensemble method. Figure 1 visualizes our approach for multilingual idiomaticity detection.[1]

4.1 Span-based Classification

There have been several attempts to utilize span hidden representations from a Transformer (Vaswani et al., 2017)-based pre-trained language model in various NLP tasks that can be formulated as span classification, including named entity recognition

[1] Appendix A describes our approaches in detail using equations.

(Yamada et al., 2020; Eberts and Ulges, 2020), relation classification (Baldini Soares et al., 2019) and propaganda technique classification (Da San Martino et al., 2020; Dimov et al., 2020; Jurkiewicz et al., 2020). These studies have all demonstrated the effectiveness of the span representations.

Here, we utilize this approach to solve the task of idiomaticity detection. We first pick up the span hidden representations of an MWE from the Transformer-based model, take their average, and feed the resulting vector into a linear layer for final classification (Figure 1 (a)).

Our concept with this approach is that the model should be able to focus more on the usage of an MWE in terms of idiomaticity in context rather than using a special [CLS] token for classification. Although we do not regard an MWE as a single token to encode, it is true that our approach is inspired by the idiom principle (Sinclair et al., 1991) in the sense that our model classifies a single averaged MWE span hidden representation for final classification.

4.2 Span Prediction-based Classification

For the second approach, we propose span prediction-based idiomaticity classification, inspired by BERT (Devlin et al., 2019)'s fine-tuning approach against the SQuAD v2.0 dataset (Rajpurkar et al., 2016, 2018), which contains some unanswerable questions. In BERT's approach, the answer text span in a given text for answerable questions is predicted, and the position of a [CLS] token for questions that do not have an answer is output. In our case, the task is to predict the MWE span in a given sequence for idiomatic MWEs and to output the position of a [CLS] token for non-idiomatic MWEs. This approach is illustrated in Figure 1 (b).

Our concept with this approach lies in the generalizability over unseen data. Predicting an idiomatic MWE span requires a model to differentiate non-idiomatic MWEs from idiomatic ones. This should force the model to learn semantic knowledge on MWEs in terms of idiomaticity and subsequently help the model to deliver a better performance on the test data.

5 Experimental Setup

Models We mainly utilized InfoXLM and XLM-R for our system submission, but we also tested several other cross-lingual pre-trained language models. Table 1 shows the list of models tested in this paper.[2] We selected these models because they are easy-to-use thanks to their availability on the HuggingFace Hub.[3]

Model	Identifier
InfoXLM (Chi et al., 2021a)	microsoft/infoxlm-large
XLM-R (Conneau et al., 2020)	xlm-roberta-large
XLM-Align (Chi et al., 2021b)	microsoft/xlm-align-base
RemBERT (Chung et al., 2021)	google/rembert
mBERT (Devlin et al., 2019)	bert-base-multilingual-cased

Table 1: List of cross-lingual pre-trained language models tested in this paper. Each identifier corresponds to the model name in the transformers library.

Data and Preprocessing We utilized the official training and development sets[4] for training, and no additional data was used, as stipulated by the competition rules. We tokenized samples using pre-trained tokenizers provided by the transformers library (Wolf et al., 2020) and set the sequence length to 256. When using an MWE as an additional input feature, we added it to the second sentence following Tayyar Madabushi et al. (2021).

Evaluation Metrics The evaluation metric for Subtask A is a macro F_1 score with respect to idiomaticity labels.

Fine-tuning We implemented our approaches using PyTorch (Paszke et al., 2019) and the transformers library. We fine-tuned all models for ten epochs each using one NVIDIA Tesla V100 (SXM2 - 32GB) with a batch size of 16 and automatic mixed precision applied. We used an Adam optimizer (Kingma and Ba, 2014) and saved a checkpoint of each model every ten steps. To minimize the effect of random seeds, we trained all models for ten times each with different random seeds. We then selected the best-performing models on the basis of the macro F_1 scores on the development set.[5]

Ensemble We fused the outputs of the fine-tuned pre-trained language models to further boost perfor-

[2] We provide a brief explanation of the five cross-lingual pre-trained language models in Appendix B and the performance comparison in Appendix E.

[3] https://huggingface.co/models

[4] https://github.com/H-TayyarMadabushi/SemEval_2022_Task2-idiomaticity

[5] For more details on hyperparameters, please refer to Appendix C.

System	Ensemble Method	Models Used
System 1	No Ensemble	InfoXLM for EN, XLM-R for PT & GL
System 2	Stacking	InfoXLM × 4, XLM-R × 1
System 3	Majority Vote	InfoXLM × 5, XLM-R × 1
System 4	Stacking	InfoXLM × 5, XLM-R × 1
System 5	Majority Vote	InfoXLM × 5, XLM-R × 1

Table 2: Configurations of our submitted systems. For ensemble methods, we used predicted labels from pre-trained language models as an input feature. For systems with stacked generalization, we trained a logistic regression model as a meta estimator.

mance on unseen data. For submission, we use either stacked generalization (Wolpert, 1992), where we train a machine learning model using predictions from pre-trained language models and corresponding idiomaticity labels and then make a final decision with it, or naïve majority voting on predicted labels. We implemented stacked generalization using scikit-learn (Pedregosa et al., 2011). Prediction labels on the development set were used as training data for a meta estimator. To train the estimator, we first divided the training data into 90% for training and 10% for hold-out. We then trained the estimator using three-fold cross-validation (CV). We tested both a ridge classifier and a logistic regression and chose the best-performing model based on the average CV score over the three validation folds. We subsequently picked up the best estimator from the resulting three models using the hold-out set. Finally, the best estimator predicted labels for the test set using the predictions of the pre-trained language models.

Submitted Systems We submitted the five models listed in Table 2 to the evaluation phase.[6] Note that all models were fine-tuned with the span-based classification approach following our preliminary experiments on the development and evaluation sets.

6 Results

Table 3 shows the official test set results for the zero-shot setting of Subtask A. Our best-performing model (System 2), using four InfoXLM models and one XLM-R model with stacked generalization, achieved a macro F_1 score of 0.7466 and was ranked fifth among 20 teams.

Ensemble We utilized ensemble methods in four out of five submissions, of which two use stacked generalization and the others adopt a naïve majority voting approach. Table 4 lists the results of our five submissions on the test set. The results indicate the effectiveness of the ensemble methods, which outperform the model with no ensemble methods by 0.0112 for the best-performing model using stacked generalization. Even for the naïve majority voting approach, the performance improved or did not fall below the result without ensembling.

Rank	Team	Macro F_1
1	clay	0.8895
2	yxb	0.8498
3	NER4ID	0.7740
4	HIT	0.7715
5	**Hitachi** (Ours)	0.7466
	Baseline	0.6540

Table 3: Top five macro F_1 scores on test set in zero-shot setting of Subtask A. Baseline uses mBERT following Tayyar Madabushi et al. (2021).

Approach		Macro F_1
No Ensemble	System 1	0.7354
Majority Vote	System 3	0.7354
	System 5	0.7448
Stacking	System 2	**0.7466**
	System 4	0.7452

Table 4: Macro F_1 test scores for our five submitted systems. All models were trained with the span-based classification approach. **Bold** indicates the best result.

Classification Approaches We verified the efficacy of three idiomaticity classification approaches—span-based classification, span prediction-based classification, and conventional `[CLS]`-based classification—using the same pre-trained model (InfoXLM). We can see in Table 5 that the span-based classification approach exhibited by far the best average macro F_1 score of 0.7303 on the test set, compared to average

[6] For the detailed configurations of each model, please see Appendix D.

Approach	Macro F_1 Development	Macro F_1 Test
Span-based	**0.7898** (.0138)	**0.7303** (.0211)
Span prediction-based	0.7514 (.0086)	0.6245 (.0255)
`[CLS]`-based	0.7166 (.0675)	0.6333 (.0371)

Table 5: Average macro F_1 development and test scores of three classification approaches with standard deviations over ten runs in parentheses. We fine-tuned InfoXLM and used the same hyperparameter settings and input features for all models.

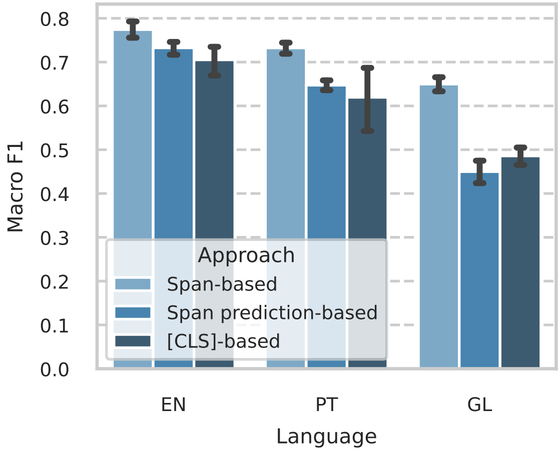


Figure 2: Average macro F_1 scores of three idiomaticity classification approaches on the test set grouped by language. Error bars denote 95% confidence interval.

macro F_1 scores of 0.6245 and 0.6333 for the span prediction-based and the `[CLS]`-based approaches, respectively. This huge difference stems partly from the Galician classification performance, since we have no associated training or development sets for Galician.

Figure 2 shows the average macro F_1 scores on the test set grouped by language. The span-based classification approach produced the highest performance across the three languages, and the performance variations among languages were relatively small, with the maximum difference of 0.1245 between English and Galician. In contrast, the span prediction-based and `[CLS]`-based approaches did not perform well on Galician samples, exhibiting average macro F_1 scores of 0.4499 and 0.4858, respectively. We assume that because idioms are generally language- and culture-specific[7] (Aldahesh, 2013; Al-kadi, 2015), it is difficult for models fine-tuned on English and Portuguese data to detect idiomaticity in unseen Galician samples without letting them know where they should be mainly looking, as in the span-based approach.

Feature	Macro F_1 Development	Macro F_1 Test
Plain	0.7859 (.0131)	0.7131 (.0196)
Plain + MWE	0.7835 (.0078)	**0.7315** (.0179)
Plain + Context	0.7898 (.0138)	0.7303 (.0211)
Plain + MWE + Context	**0.7918** (.0141)	0.7280 (.0212)

Table 6: Average macro F_1 development and test scores with standard deviations over ten runs in parentheses. “Plain” denotes a target sentence, while “Context” represents the previous and next sentences. We fine-tuned InfoXLM using the span-based classification approach and used the same hyperparameter settings for all models.

Input Features Tayyar Madabushi et al. (2021) reported that encoding a target sentence along with its adjacent contexts showed the best classification performance in the zero-shot setting among the four possible input feature combinations: (i) a target sentence, (ii) a target sentence with its MWE as a second sentence, (iii) a target sentence with its adjacent contexts, and (iv) a target sentence, its MWE and adjacent contexts. Here, we also investigated these combinations using the span-based classification approach (Table 6). The results indicate that considering a target sentence and its adjacent contexts is not always the best option. In our experiments, utilizing a target sentence and its target MWE as inputs (Plain + MWE) achieved the best average macro F_1 score of 0.7315, followed by Plain + Context with a macro F_1 score of 0.7303. While using only a target sentence showed comparable performance to the other approaches on the development set, it ended up producing the worst result on the test set. These results suggest that using an additional feature along with a target sentence is likely to boost detection performance, but it is not clear which combination of input features yields the best performance given the standard deviations.

7 Conclusion

In this paper, we have proposed two approaches for detecting idiomaticity in a given sequence: span-based classification and span prediction-based classification. While the performance of the latter was almost on par with that of the well-known standard sequence classification approach using a `[CLS]`

[7] Although Portuguese and Galician have strong historical ties, they are categorized as two different languages (Ramallo and Rei-Doval, 2015; Garcia, 2021).

hidden representation, the former outperformed it and showed the best macro F_1 score of 0.7466, which ranked fifth in the zero-shot setting of Subtask A. We also found that it is essential to guide a model on which tokens to look at when no training data is available for a particular language. In future work, we will investigate a more effective idiomaticity detection approach against unseen language data.

Acknowledgements

We would like to thank anonymous reviewers for their valuable feedback and Dr. Masaaki Shimizu for maintenance and management of large computational resources.

References

Abdu Mohammad Talib Al-kadi. 2015. Towards idiomatic competence of yemeni efl undergraduates. *Journal of Language Teaching and Research*, 6(3):513.

Ali Yunis Aldahesh. 2013. On idiomaticity in english and arabic: A cross-linguistic study. *Journal of Languages and Culture*, 4(2):23–29.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Andrea Cascallar-Fuentes, Alejandro Ramos-Soto, and Alberto Bugarín Diz. 2018. Adapting SimpleNLG to Galician language. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 67–72, Tilburg University, The Netherlands. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. Improving pretrained cross-lingual language models via self-labeled word alignment. 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 3418–3430, Online. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

Alexis Conneau and Guillaume Lample. 2019. *Cross-Lingual Language Model Pretraining*. Curran Associates Inc., Red Hook, NY, USA.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Dimov, Vladislav Korzun, and Ivan Smurov. 2020. NoPropaganda at SemEval-2020 task 11: A borrowed approach to sequence tagging and text classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1488–1494, Barcelona (online). International Committee for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. *ArXiv*, abs/1909.07755.

Marcos Garcia. 2021. Exploring the representation of word meanings in context: A case study on homonymy and synonymy. 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 3625–3640, Online. Association for Computational Linguistics.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. 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.

Reyhaneh Hashempour and Aline Villavicencio. 2020. Leveraging contextual embeddings and idiom principle for detecting idiomaticity in potentially idiomatic expressions. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 72–80, Online. Association for Computational Linguistics.

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. ApplicaAI at SemEval-2020 task 11: On RoBERTa-CRF, span CLS and whether self-training helps them. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1415–1424, Barcelona (online). International Committee for Computational Linguistics.

Milton King and Paul Cook. 2018. Leveraging distributed representations and lexico-syntactic fixedness for token-level prediction of the idiomaticity of English verb-noun combinations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 345–350, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Francis Jeffry Pelletier. 1994. The principle of semantic compositionality. *Topoi*, 13(1):11–24.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Fernando Ramallo and Gabriel Rei-Doval. 2015. The standardization of galician. *Sociolinguistica*, 29(1):61–82.

Giancarlo Salton, Robert Ross, and John Kelleher. 2016. Idiom token classification using sentential distributed semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 194–204, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

J. Sinclair, L. Sinclair, and R. Carter. 1991. *Corpus, Concordance, Collocation*. Describing English language. Oxford University Press.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Appendices

A Approaches

Here, we describe our two approaches in detail using mathematical notations.

A.1 Span-based Classification

Given $\mathcal{S}$, $\mathcal{W}$, and y_{MWE}, we first tokenize $\mathcal{S}$ and $\mathcal{W}_{\text{MWE}}$ using a pre-trained tokenizer and obtain their token-level representations: $\mathcal{S}' = [t_1, \ldots, t_{n' \in \mathbb{N}}]$ and $\mathcal{W}' = \left[t_1^{\text{MWE}}, \ldots, t_{m' \in \mathbb{N}}^{\text{MWE}}\right]$. We then feed $\mathcal{S}'$ into a Transformer-based pre-trained language model and obtain the output hidden representation $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n']$. We pick up only the hidden representation of $\mathcal{W}'$ and compute their average as $\mathbf{h}_{\text{MWE}} = \frac{1}{|\mathcal{W}'|}\left(\mathbf{h}_1^{\text{MWE}} + \cdots + \mathbf{h}_{m'}^{\text{MWE}}\right)$. Finally, we put $\mathbf{h}_{\text{MWE}}$ into an output linear layer and obtain the prediction. The training objective in this task is the binary cross-entropy loss.

A.2 Span Prediction-based Classification

Given $\mathcal{S}'$, a start position of $\mathcal{W}'$ and an end position of $\mathcal{W}'$, we first feed $\mathcal{S}$ into a Transformer-based model and then put the output representation $\mathbf{H}$ to a linear classifier for classification, yielding $\mathbf{O} \in \mathbb{R}^{n' \times 2} = [\mathbf{o}_{\text{start}}; \mathbf{o}_{\text{end}}]$. We finally apply the softmax function to $\mathbf{o}_{\text{start}}$ and $\mathbf{o}_{\text{end}}$ in order to obtain the idiomatic MWE span probabilities. For prediction, we first calculate the maximum scoring span and obtain its score as $s = o_i^{\text{start}} + o_j^{\text{end}}$, where j must be greater than i, and o_i^{start} and o_j^{end} are the i-th and j-th values of $\mathbf{o}_{\text{start}}$ and $\mathbf{o}_{\text{end}}$, respectively. We also calculate the non-idiomatic score as $\overline{s} = o_1^{\text{start}} + o_1^{\text{end}}$, where index 1 refers to the index of the `[CLS]` token. If $s \geq \overline{s}$, $\mathcal{W}$ is regarded as *idiomatic*. Otherwise, $\mathcal{W}$ is predicted as a non-idiomatic MWE. This task is trained with an average of the log-likelihoods of the correct start $\mathcal{L}_{\text{start}}$ and end $\mathcal{L}_{\text{end}}$ positions: $\mathcal{L}_{\text{span}} = \frac{1}{2}\left(\mathcal{L}_{\text{start}} + \mathcal{L}_{\text{end}}\right)$.

B Cross-lingual Pre-trained Language Models

We briefly explain the five cross-lingual pre-trained language models tested in this paper.

- mBERT (Devlin et al., 2019): Pre-trained with multilingual masked language modeling using Wikipedia. Its architecture follows that of BERT-BASE.

- XLM-R (Conneau et al., 2020): Pre-trained with multilingual masked language modeling using CommomCrawl, which is much larger than Wikipedia. The architecture generally follows that of BERT-LARGE.

- InfoXLM (Chi et al., 2021a): Pre-trained with multilingual masked language modeling, translation language modeling, and the newly proposed cross-lingual contrastive learning, using CommonCrawl. The architecture follows that of XLM-R.

- XLM-Align (Chi et al., 2021b): Pre-trained with multilingual masked language modeling, translation language modeling and denoising word alignment, using CommonCrawl and

- Wikipedia. The architecture generally follows that of BERT-BASE.

- RemBERT (Chung et al., 2021): Pre-trained with multilingual masked language modeling using both CommonCrawl and Wikipedia. The architecture is completely different from that of XLM-R, though it has the same number of parameters (559M). It consists of 32 hidden layers, 18 attention heads and $Dim_{\text{hidden}} = 1152$.

Note that all models can accommodate the three target languages (English, Portuguese, and Galician).

C Hyperparameter Settings

Table 8 shows the hyperparameter settings. We explored various hyperparameter combinations with respect to a pre-trained language model, a peak learning rate, and an input feature and selected the models with a macro F_1 score of 0.795 or above on the development set. Note that we also tested XLM-Align, RemBERT, and mBERT in our preliminary experiments, but these did not perform well on the development set (see Appendix E); therefore, we did not use them in our submissions.

D Model Configurations

Table 9 lists the models used for our submissions, while Table 10 shows the configurations of our five submitted systems. For System 1, we used the two different best-performing models for English and Portuguese.[8] For Galician, because we did not have any training samples provided in the zero-shot setting, we used the same model as Portuguese, as both Galician and Portuguese have grammatical and lexical similarities due to their shared historical background (Ramallo and Rei-Doval, 2015; Cascallar-Fuentes et al., 2018; Garcia, 2021).

E Performance Comparison of Five Cross-lingual Pre-trained Language Models

Table 7 compares average macro F_1 scores of five cross-lingual pre-trained language models on the development and test sets. Interestingly, RemBERT produced the best result on the test set with an average macro F_1 score of 0.7452, though it ranked third for the development set. This is presumably because the Galician and English classification performances of RemBERT are better than any other cross-lingual pre-trained language models that we tested (Figure 3).

Model	**Macro F_1**	
	Development	Test
InfoXLM	0.7898 (.0139)	0.7304 (.0211)
XLM-R	**0.7959** (.0110)	0.7116 (.0125)
XLM-Align	0.7600 (.0096)	0.7015 (.0119)
RemBERT	0.7833 (.0090)	**0.7452** (.0192)
mBERT	0.7440 (.0125)	0.7014 (.0125)

Table 7: Average macro F_1 development and test scores of five cross-lingual pre-trained language models with standard deviations over ten runs in parentheses. We use the same hyperparameter settings for all models.

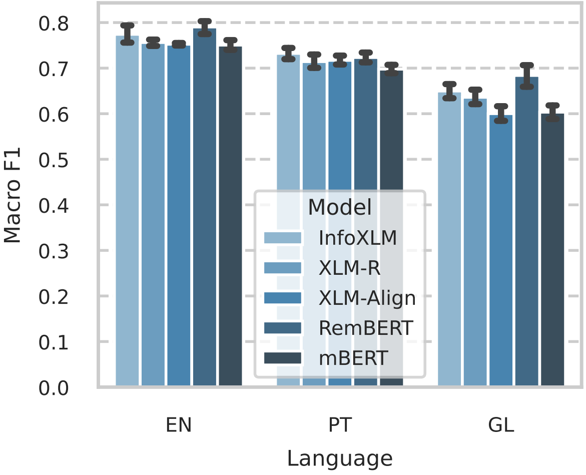


Figure 3: Average macro F_1 scores of five cross-lingual pre-trained language models on the test set grouped by language. Error bars denote 95% confidence interval.

[8] We selected the best-performing models based on macro F_1 scores on the evaluation set.

Hyperparameter	**Candidate**
Batch size	16
Epochs	10
Model	(InfoXLM, XLM-R)
Peak learning rate	(0.5e-5, 1e-5, 1.5e-5, 2e-5, 2.5e-5, 3e-5)
Input feature	(Plain + MWE, Plain + Context, Plain + MWE + Context)
Warmup steps	5% of steps
Weight decay	0.01
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999
Sequence length	256
Attention Dropout	0.1
Dropout	0.1

Table 8: Hyperparameters in our experiments. We explored various hyperparameter combinations with respect to pre-trained language model, peak learning rate, and input feature. If not specifically mentioned in the paper, we used hyperparameters denoted with an underline.

	Model Type	**Hyperparameter**			**Macro F$_1$**					
					Development			Evaluation		
		LR	Input Feature	Seed	EN	PT	All	EN	PT	All
$\mathcal{M}_1$	InfoXLM	2e-5	Plain + Context	25	.800	**.814**	.814	**.862**	.678	.801
$\mathcal{M}_2$	InfoXLM	1e-5	Plain + MWE	25	.787	.796	.799	.858	.667	.797
$\mathcal{M}_3$	XLM-R	1e-5	Plain + Context	42	.814	.746	.797	.850	**.743**	**.817**
$\mathcal{M}_4$	InfoXLM	1e-5	Plain + Context	42	.799	.770	.797	.844	.679	.792
$\mathcal{M}_5$	InfoXLM	1e-5	Plain + MWE + Context	22	.788	.784	.796	.854	.696	.803
$\mathcal{M}_6$	InfoXLM	1.5e-5	Plain + Context	42	**.837**	.742	.810	.800	.665	.762
$\mathcal{M}_7$	InfoXLM	2e-5	Plain + MWE + Context	29	.808	.806	**.818**	.854	.708	.812

Table 9: List of models used in our submissions and their macro F$_1$ scores on the development and evaluation sets. **Bold** indicates the best result in each category, while underline indicates the second-best result.

System	**Approach**	**Models Used**	**Macro F$_1$**			
			EN	PT	GL	All
System 1	No Ensemble	$\mathcal{M}_1$ for EN, $\mathcal{M}_3$ for PT & GL	**.820**	.733	.614	.735
System 2	Stacking	$\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5$	.783	**.761**	.663	**.747**
System 3	Majority Vote	$\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_6$	.785	.739	.647	.735
System 4	Stacking	$\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_7$	.785	.757	.660	.745
System 5	Majority Vote	$\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_7$	.769	.753	**.685**	.745

Table 10: Configurations of our submitted systems and their macro F$_1$ scores on the test set. **Bold** indicates the best result in each category, while underline indicates the second-best result.