BERT-based Idiom Identification using Language Translation and Word Cohesion

Arnav Yayavaram*, Siddharth Yayavaram*, Prajna Upadhyay, Apurba Das
BITS Pilani Hyderabad, India
{f20213117, f20213116, prajna.u, apurba}@hyderabad.bits-pilani.ac.in

Abstract

An idiom refers to a special type of multi-word expression whose meaning is figurative and cannot be deduced from the literal interpretation of its components. Idioms are prevalent in almost all languages and text genres, necessitating explicit handling by comprehensive NLP systems. Such phrases are referred to as Potentially Idiomatic Expressions (PIEs) and automatically identifying them in text is a challenging task. In this paper, we propose using a BERT-based model fine-tuned with custom objectives, to improve the accuracy of detecting idioms in text. Our custom loss functions capture two important properties (word cohesion and language translation) to distinguish PIEs from non-PIEs. We conducted several experiments on 7 datasets and showed that incorporating custom objectives while training the model leads to substantial gains. Our models trained using this approach also have better sequence accuracy over DISC, a state-of-the-art PIE detection technique, along with good transfer capabilities. Our code and datasets can be downloaded from https://github.com/siddharthyayavaram/BERT-Based-Idiom-Detection

Keywords: idioms, multi-word expressions, word cohesion, language translation, loss function

1. Introduction

An idiom refers to a special type of multi-word expression (Baldwin and Kim, 2010) whose meaning is figurative and cannot be deduced from the literal interpretation of its components. Idioms often exhibit peculiar behavior by violating selection restrictions or altering the default semantic roles of syntactic categories. Consequently, they pose significant challenges for Natural Language Processing (NLP) systems. Idioms are prevalent in almost all languages and text genres, necessitating explicit handling by comprehensive NLP systems. We refer to these phrases as *potentially idiomatic expressions (PIEs)* to account for the contextual semantic ambiguity in their expression. Better detection of PIEs can enhance numerous machine translation tasks.

Techniques to automatically detect and identify PIEs need to do many tasks accurately – $i)$ automatically detect if an idiomatic expression is present in a sentence (Briskilal and Subalalitha, 2022; Tan and Jiang, 2021; Liu and Hwa, 2019), $ii)$ if yes, identify the idiomatic tokens (Zeng and Bhat, 2021, 2022). Both of these are challenging tasks. For instance, in the sentence"Oh — for about four years, on and off, he said vaguely", the potentially idiomatic expression "on and off" is used figuratively, whereas, it is used literally in the sentence "Participate in training, both on and off station". Existing techniques for idiom detection rely on syntactic patterns, knowing the PIE being classified correctly, and lack generalization. In this paper, we address the above-mentioned problems and show that improvement in $i)$ improves $ii)$ substantially.

We employ a BERT-based fine-tuning approach with custom objectives to improve accuracy on all 3 tasks. We define our objectives in Section 4.2 based on language translation and word cohesion.

Our salient contributions are:

1: Introduction of a language translation-based metric to detect the presence of idioms.

2: A novel loss function to selectively penalize examples using sentence translation and word cohesion that can be used with any architecture for idiom detection.

3: Our models trained with custom loss functions exhibit improved generalization capabilities, evident in identifying unseen PIEs.

2. Related Work

MWE, short for Multi Word Expressions are notable collocations with multiple words, for instance "all at once" or "look something up". (Baldwin and Kim, 2010; Constant et al., 2017). IEs (Idiomatic Expressions), are a subset of MWEs, which exhibit non-compositionality (Baldwin and Kim, 2010; Fadaee et al., 2018; Liu et al., 2017; Biddle et al., 2020). Metaphors, such as "heart of gold" and "night owl" compare unrelated things implicitly. While some MWEs and IEs use metaphorical figuration, not all metaphors are IEs; they can be direct comparisons with single words (e.g., "I am titanium"). In this paper, we study IEs.

*Equal Contribution

Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD 2024)
@*LREC-COLING-2024*, pages 220–230. May 25, 2024.

IE Classification broadly falls under two categories – standalone phrase classification and context-based classification. Standalone classification tasks decide if a phrase could be used as an idiom without specifically considering its context (Fazly and Stevenson, 2006; Shutova et al., 2010; Tabossi et al., 2008, 2009; Reddy et al., 2011; Cordeiro et al., 2016) as opposed to context-based idiom classification techniques which take into account the entire sentence to detect the presence of idiom (Peng et al., 2014; Nedumpozhimana et al., 2022; Peng and Feldman, 2017; Tan and Jiang, 2021; Verma and Vuppuluri, 2015; Briskilal and Subalalitha, 2022; Liu and Hwa, 2019). Earliest known context-based phrase classification techniques developed per idiom classifiers, which are not scalable (Liu and Hwa, 2017). Context-based phrase classification techniques can additionally detect which tokens are idiomatic/nonidiomatic (Zeng and Bhat, 2021; Salton et al., 2016; Zeng and Bhat, 2022). Typically, the latter is dependent on the former task – only if an idiom is detected to be present in a sentence, does the classification of idiomatic and non-idiomatic tokens follow. Efforts to build complementary resources to support this task include constructing a knowledge graph (Zeng et al., 2023) and an information retrieval system to search for idiomatic expressions (Hughes et al., 2021).

Detecting idioms in the text has also become popular in non-English languages. In (Itkonen et al., 2022), authors leverage various models provided by HuggingFace in conjunction with the standard BERT model for the idiom detection task in English, Portuguese, and Galician. They emphasize on feature engineering using traits that define idiomatic expressions. These additional features result in enhancements compared to the baseline performance. In (Tedeschi et al., 2022), a multilingual transformer based model and a dataset of idioms in 10 languages is presented. A rule-based intra-sentential idiom detection system in Hindi was presented in (Priyanka and Sinha, 2014).

3. Problem Statement

We are given the following:

- A sentence $\mathcal{S}$ with n tokens $w_1, w_2, \ldots, w_n$, where each w_i represents a tokenized unit. $\mathcal{S}$ is an syntactic ordering over w_i's.
- Labels $\mathcal{L} = \{\texttt{I}, \texttt{NI}\}$ where `I` and `NI` represent `<idiom>` and `<not idiom>` (or literal) classes, respectively.

This labelling produces a sequence of class labels $Z = z_1, z_2, \ldots, z_n$ where $z_i = f(w_i)$. The high-level objective of this work is to learn the function $f(\cdot)$

- A successful prediction occurs when an idiomatic subsequence $w_{i:j}$ is identified in S, and the corresponding labels $z_{i:j}$ are labelled as `I`. There can be more than one such subsequences.
- If the subsequence $w_{i:j}$ is literal, all corresponding labels $z_{i:j}$ are `NI`.
- If the sentence lacks an idiom, all $z_{1:n}$ are categorized as `NI`.

4. Methodology

4.1. BERT-based Idiom Identification

Figure 1 shows the high-level architecture of our method. Our loss functions are implemented over BERT (Devlin et al., 2018), a pre-trained transformer-based model developed by Google. Due to its effectiveness in capturing context and semantics for various NLP tasks, we re-use its pre-trained architecture for fine-tuning our model using binary cross-entropy loss. Despite its success, cross entropy loss is sensitive to outliers and class-imbalance. We observe class imbalance in idiom classification where the label `I` is far less frequent than label `NI` leading to poor accuracy for `I` tokens. To fix this, we propose to use language translation and word cohesion to manipulate the loss. In the following sections, we define two novel loss functions for the task of idiom token classification. The merit of our work lies in the fact that these custom loss functions can be used with **any** architecture.

4.2. Language Translation and Cohesion for Idioms

4.2.1. Translation-based Loss Function

An important property exhibited by an idiom is the difference between its literal and actual meaning. However, a phrase that is an idiom in language L_1 is improbable to be an idiomatic phrase in another language L_2.For example, take the English idiom, "raining cats and dogs", its Hindi translation is "भारी वर्षा", which when translated back to English gives "heavy rain" which is the meaning of our initial idiom but is quite different lexically. Let $\mathcal{S}_{L_1}$ denote a sentence containing an idiom in language L_1, $\mathcal{S}_{L_1 \to L_2}$ a translation of $\mathcal{S}_{L_1}$ in L_2, and $\mathcal{S}_{L_1 \rightleftarrows L_2}$ a translation of $\mathcal{S}_{L_1 \to L_2}$ back to L_1. When $\mathcal{S}_{L_1}$ is translated to $\mathcal{S}_{L_1 \to L_2}$, the idiomatic tokens in $\mathcal{S}_{L_1}$ will be expressed through their actual meaning in $\mathcal{S}_{L_1 \to L_2}$ because of a lack of corresponding idiom in L_2. Re-translating it to L_1 will force the idiom to be expressed with its actual meaning in $\mathcal{S}_{L_1 \rightleftarrows L_2}$. Lexically, the actual meaning of an idiom and the surface form of an idiom differ substantially from each other. We employ this simple trick to detect

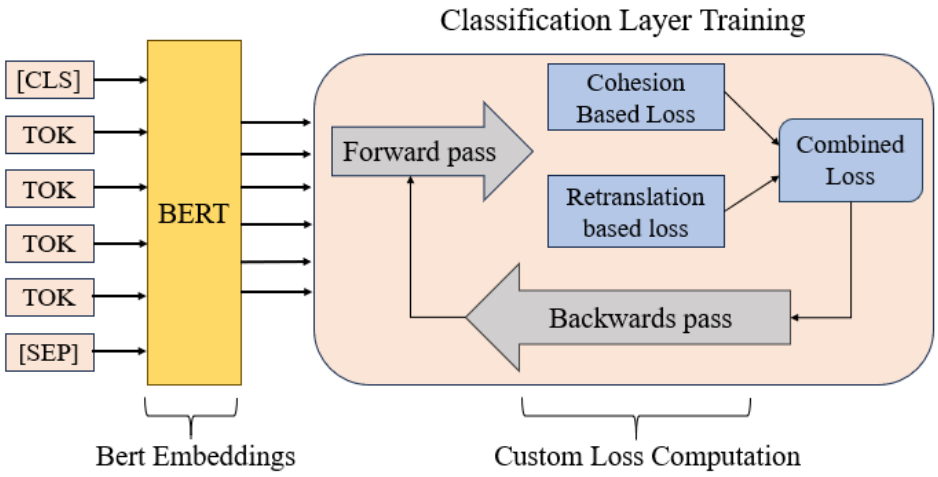

Figure 1: Architecture of our proposed method

the presence of an idiom in a sentence – if $\mathcal{S}_{L_1 \rightleftarrows L_2}$ and $\mathcal{S}_{L_1}$ differ lexically by some margin, $\mathcal{S}$ is likely to contain an idiom. A sentence that contains no idiom is likely to have the same lexical representation in the original and back-translated sentence. We leverage the METEOR (Banerjee and Lavie, 2005) metric to quantify this observation by computing a score to reflect the lexical and syntactic similarity between the translated and reference sentences. METEOR incorporates a penalty mechanism for longer matches by organizing system translation unigrams mapped to reference translation unigrams into minimal chunks. These chunks consist of adjacent unigrams in the system translation that align with adjacent unigrams in the reference translation. Longer n-grams result in fewer chunks. In the extreme case of a complete match, only one chunk exists, while in the absence of bigram or longer matches, the number of chunks equals the count of unigram matches. An alignment is created between the system translation and the reference translation by mapping unigrams based on different criteria, such as exact match, stemming, or synonymy. The alignment is formed by selecting the most extensive subset of unigram mappings, ensuring that each unigram maps to at most one unigram in the other string. The chosen alignment is the one with the fewest "unigram mapping crosses", which occur when lines connecting mapped unigrams intersect in a vertical arrangement of the two strings.

$$\text{Unigram Precision:} \quad \mathcal{P} = \frac{N_{\text{correct}}}{N_{\text{backtrans}}}$$

$$\text{Unigram Recall:} \quad \mathcal{R} = \frac{N_{\text{correct}}}{N_{\text{original}}}$$

Here, N_{correct} represents the number of correctly mapped unigrams, $N_{\text{backtrans}}$ represents the total number of unigrams in the back-translated sentence, and N_{original} represents the total number of unigrams in the original sentence.

$$\text{Harmonic Mean:} \quad \mathcal{F}_{\text{mean}} = \frac{10 \cdot \mathcal{P} \cdot \mathcal{R}}{\mathcal{R} + 9 \cdot \mathcal{P}}$$

$$\text{Penalty} = 0.5 \times \left(\frac{C}{U}\right)^3$$

where C represents the number of chunks and U represents the number of unigrams matched.

$$\text{Score} = \mathcal{F}_{\text{mean}} \times (1 - \text{Penalty})$$

It evaluates the quality of a translation by comparing it to one or more reference translations. METEOR considers various factors such as unigram precision, recall, and alignment errors to compute a score that reflects the lexical and syntactic similarity between the translated and reference sentences. For instance, the sentence "The early morning flight required them to hit the sack much earlier than usual", is translated into Italian "Il volo mattutino li obbligava a coricarsi molto prima del solito.", and its back-translation to English "The morning flight forced them to go to bed much earlier than usual.", the idiomatic usage causes a large syntactic change during back-translation which will lead to a high alignment error term and comparatively lower METEOR score of 0.5919.

During the training of the BERT-based model for idiom recognition, the translation-based loss function incorporates the METEOR score as a penalty term. If the METEOR score falls below a certain threshold, it indicates that the back-translation process has significantly altered the original sentence, which we posit is due to the presence of idiomatic expressions.

$$\mathcal{L}_{retranslation} = \mathcal{L}(1 + \lambda_1 \mathbb{1}(\mathcal{MS} < \lambda_2)) \quad (1)$$

where $\mathcal{MS}$ is the meteor score for the sentence, $\mathcal{L}$ is the original binary cross entropy loss, and $\mathbb{1}(|\mathcal{MS}| < \lambda_2)$ is an indicator function. It takes a value of 1 if it is low ($< \lambda_2$) which scales the loss λ_1 times. Otherwise, it defaults to regular loss $\mathcal{L}$.

By increasing the loss for examples where idioms are not accurately retained through back-translation, the model is encouraged to better understand and retain the meaning of idiomatic expressions. This, in turn, leads to improved performance metrics such as precision and recall, as the model becomes more adept at recognizing and appropriately handling idiomatic language during inference, resulting in better generalization to unseen data.

4.2.2. Cohesion based Loss Function

Idioms exhibit a lack of semantic compositionality or *cohesion* among its words also reported in earlier work (Baldwin and Kim, 2010). Given a sentence $\mathcal{S}$ where all tokens in the subsequence $w_{i:j}$ are tagged as `I`, we quantify the cohesion $C_{\mathcal{S}}$ among the words in $\mathcal{S}$ using Equation 2. It captures the mean similarity among the words in $\mathcal{S}$.

$$C_{\mathcal{S}} = \frac{1}{N} \sum_{w_i, w_j \in \mathcal{S}, i \neq j} \text{sim}(V(w_i), V(w_j)) \quad (2)$$

where $V(w_i)$ is an embedding vector for w_i, N is the total number of pairs of tokens in $\mathcal{S}$, and sim$(V(w_i), V(w_j))$ captures semantic similarity between w_i and w_j using $V(w_i)$ and $V(w_j)$. The 'sim' score is computed as the cosine similarity between the high dimensional vectors for each word. Its values range from -1 to 1, where 1 indicates high similarity and lexical cohesion, 0 represents dissimilar or orthogonal tokens, and -1 suggests that the vectors are in opposite directions. Similarly, we compute $C_{\mathcal{S}'}$, where $\mathcal{S}'$ is a sentence with the idiom tokens $w_{i:j}$ removed. The key idea is if $C_{\mathcal{S}'}$ is substantially higher than $C_{\mathcal{S}}$, then the $\mathcal{S}$ is highly likely to contain an idiomatic phrase. This follows from the intuition that idiomatic tokens are remotely related semantically to non-idiomatic tokens in $\mathcal{S}$ and their removal should increase the cohesion score.

We introduce this idea as loss during the fine-tuning objective. By penalizing examples with `I` classifications that are not likely to contain idioms, it is guiding the model to differentiate between idiomatic and non-idiomatic sentences. Our cohesion-based loss function $\mathcal{L}_{cohesion}$ is expressed in Equation 3.

$$\mathcal{L}_{cohesion} = \mathcal{L}(1 + \lambda_3 \mathbb{1}(|\mathcal{C}_{S_1} - \mathcal{C}_{S_2}| > \lambda_4)) \quad (3)$$

where $\mathcal{C}_{S_1}$ and $\mathcal{C}_{S_2}$ are the cohesion scores for sentence $\mathcal{S}$ without and with the target idiom, respectively, $\mathcal{L}$ is the original binary cross entropy loss, and $\mathbb{1}(|\mathcal{C}_{S_1} - \mathcal{C}_{S_2}| > \lambda_4)$ is an indicator function. It takes a value of 1 if there is sufficient difference between cohesion scores $\mathcal{C}_{S_1}$ and $\mathcal{C}_{S_2}$ ($> \lambda_4$) which scales the loss λ_3 times. Otherwise, it defaults to regular loss $\mathcal{L}$.

4.3. Final Loss

The final loss is a linear combination of $\mathcal{L}_{retranslation}$ and $\mathcal{L}_{cohesion}$.

$$\mathcal{L}_{final} = \tau_1 \mathcal{L}_{retranslation} + \tau_2 \mathcal{L}_{cohesion} \quad (4)$$

τ_1 and τ_2 ($0 \leq \tau_i \leq 1$) are parameters to control the effect of both losses. These parameters depend on the accuracy of $C_{\mathcal{S}}$ and $\mathcal{MS}$, which is determined by the quality of underlying embedding vectors (Equation 2) and translation API used. More weight can be given to the more accurate value.

5. Experiments

In this section, we present an empirical evaluation of our models on synthetic and real-world datasets to show the capabilities of our custom loss functions. We also compare our models with state-of-the-art techniques like DISC (Zeng and Bhat, 2021) — and we observed that using our custom loss functions leads to improved accuracies.

5.1. Experimental Setup

For training and testing our models, we make use of a 32×2 cores AMD EPYC5037532 server with 1 TB of RAM, and 8x A100 SXM4 80GB504. We used `bert-based-uncased` as our base model which we finetune.

In our experiments, we adapted the pre-trained `bert-base-uncased` (Devlin et al., 2018) model from Hugging Face [1] and proceed with fine-tuning. We selected this model primarily for its moderate size, which strikes a balance between performance and computational efficiency. Additionally, the "uncased" variant simplifies text processing by disregarding case sensitivity, making it faster to process. These factors make it a practical choice for token classification tasks without compromising performance. We selected Hindi as the language we translate to.

We partitioned each dataset into training (80%), validation (10%), and test sets (10%). Next, we applied a BERT tokenizer on the texts for generating tokens. This step is essential because it transforms the raw text data for input into the BERT model, which operates at the token level rather

[1] https://huggingface.co/docs/trl/en/models

than the word level. By converting words into token IDs, the tokenizer enables the model to understand and process the text effectively.

After tokenization, we aligned the labels with tokens to establish the correspondence between input tokens and the corresponding class labels. This alignment ensures that each token in the input text is associated with the correct entity label, allowing the model to learn the mapping between tokens and entity types during training. The alignment function handles cases where words are split into subwords by the tokenizer, ensuring that the labels are assigned appropriately to each token, even in the presence of subwords. We excluded special tokens representing separation between sentences and the start of the sentence from the training loss calculation by assigning them special labels.

We trained our model for three epochs, observing a sharp drop in loss over each epoch with a learning rate of '2e-5'. The training and evaluation batch sizes were set to 16. Weight decay was set to '0.01' to avoid overfitting. We set λ_1, λ_2, λ_3, and λ_4 all to 999, and τ_1 and τ_2 to 0.01. We repeated our experiments for three seeds and reported average accuracy values (Table 2). Additionally, it is worth noting that we observe minimal deviation in accuracy across different random seeds which underscores the robustness of the results.

5.2. Baselines

BERT-based approach (without custom loss). We fine-tuned the BERT model with binary cross-entropy loss.

BERT-based approach (with loss). We used translation, cohesion, and combination losses (described in Section 4.2) to fine-tune our BERT model.

DISC. The DISC model is based on BERT, it uses contextualized and static embeddings to encode tokens using attention, and performs token-level literal/idiomatic classification, resulting in the final output. We compare DISC with our models on the Sequence Accuracy metric described in Section 5.4.

5.3. Datasets

Table 1 describes statistics of all the datasets we have used.

1) magpie. Derived from the British National Corpus (BNC) and annotated for idiomatic expressions (PIEs)(Haagsma et al., 2020)(Consortium, 2007), the MAGPIE corpus comprises 1756 PIEs across various syntactic patterns, alongside 56622 annotated instances (32.24 per PIE). We focused on fully figurative or literal samples, ensuring unambiguous tagging reflected in confidence scores. The resulting dataset includes approximately 37000 complete sentences, excluding those longer than 50 tokens.

Dataset	total number of sentences	#idioms	#sentences containing idioms	average sentences per idiom
MAGPIE	36192	1727	27727	16.05
VNC-Tokens	2571	48	2111	43.97
theidioms	7380	1606	7830	4.87
formal	3136	358	3136	8.76
gtrans	440	22	440	20
gpt+gtrans	880	22	440	20
theidioms 1-1	1606	1606	1606	1

Table 1: Statistics of the datasets used

2) VNC-Tokens Dataset. The VNC (Verb-Noun Combinations) corpus, sourced from the British National Corpus (BNC)(Cook et al., 2008)(Consortium, 2007), comprises 53 potentially idiomatic expressions (PIEs) with about 2500 annotated sentences, categorized as literal or figurative. Using regular expression libraries and the NLTK library [2], we annotated tokens as idiomatic or non-idiomatic, leveraging prior knowledge of the idiomatic expressions for pattern matching(Cook et al., 2008) .

3) theidioms. We scraped 1606 of the most common English idioms from theidioms.com website using the Beautiful Soup library, resulting in a dataset of 7830 sentences. A few example sentences accompany each idiom. We use the NLTK library for lemmatization and text processing. We used a function to identify positions in sentences where a phrase similar to the idiomatic phrase occurs based on the lemmatized tokens and a similarity threshold. We use a similarity threshold of 0.9, ensuring that even slight variations of the idiomatic phrases are selected and annotated, as the idioms in the example sentences do not maintain the same format across all examples or instances of its usage. We have released a file containing the unfiltered sentences corresponding to particular idiomatic expressions.

4) formal. We utilized the EPIE corpus (English Possible Idiomatic Expressions)(Saxena and Paul, 2020), consisting of 25027 sentences. The corpus is divided into Formal and Static idioms, with 3136 sentences containing 358 Formal idioms and 21891 sentences containing 359 Static idioms. Static idioms are expressed using the exact phrase

[2] https://www.nltk.org/

in all sentences, whereas formal idioms undergo lexical changes across instances. The token labeling follows the BIO convention with tags `B-IDIOM` (beginning of PIE), `I-IDIOM` (continuation of PIE), and `O` (Non-Idiom token). We merged `B-IDIOM` and `I-IDIOM` into one token to match our other datasets and treat this problem as a binary token classification task. We only focus on the formal portion of this dataset as the lexical changes to the expressions address a more robust task.

5) gtrans. We compiled a dataset of 440 sentences using GPT-3.5, featuring 22 English idioms sourced manually from online platforms. Each idiom was paired with 20 example sentences. After translating these idioms to Hindi and then back to English, we observed that Google Translate accurately retained their meanings, demonstrating its understanding of these idioms.

6) gpt+gtrans. We added 440 sentences generated by GPT-3.5 without idiomatic expressions to the `gtrans` dataset, resulting in a total of 880 sentences. 440 with idiomatic expressions present, and 440 without idioms. Token labeling and annotation followed similar methods as in previous datasets. Additionally the sentences without idioms have all tokens labeled as 0.

7) theidioms 1-1. The dataset, sourced from `theidioms.com`, contains 1606 idioms (also present in `theidioms`), each with a single instance, ensuring a 1-1 mapping between sentences and idioms. We labeled tokens using pattern matching and text processing with the NLTK library. This dataset tests the model's generalization by including idioms unseen during training.

5.4. Metrics

Precision, Recall, F1. We calculated precision, recall, and F1-scores for both `I` and `NI` classes, presenting them as ordered pairs.

Macro and Weighted Average F1. We calculated macro average as a mean of the values of the ordered pair, and the weighted average considering the relative number of each token in the complete dataset.

Weighted-Averaged Formulae

$$P = \frac{\sum_{i=1}^{N}(TP_i + FP_i) \times P_i}{\sum_{i=1}^{N}(TP_i + FP_i)}$$

$$R = \frac{\sum_{i=1}^{N}(TP_i + FN_i) \times R_i}{\sum_{i=1}^{N}(TP_i + FN_i)}$$

$$F1 - \text{score} = \frac{\sum_{i=1}^{N}(2 \times P_i \times R_i) \times (TP_i + FN_i)}{\sum_{i=1}^{N}(P_i + R_i) \times (TP_i + FN_i)}$$

Where **P**: Precision; $\mathbf{P_i}$: Precision of the i^{th} example; **R**: Recall; $\mathbf{R_i}$: Recall of the i^{th} example; **N**: Number of classes (2 in our case); $\mathbf{TP_i}$: True Positives for class i; $\mathbf{FP_i}$: False Positives for class i; $\mathbf{FN_i}$: False Negatives for class i; $\mathbf{TN_i}$: True Negatives for class i.

Sequence Accuracy. A sentence is only considered correct if all of its constituent tokens are correctly marked. This metric can be considered as a much more stringent metric than normal F1 and accuracy scores (Zeng and Bhat, 2021).

5.5. Results

5.5.1. With Regular Loss

Table 2 shows our results. Our base models utilizing regular binary cross entropy loss display good baseline results, however the results are consistently the lowest across all datasets and experiments compared to using custom loss functions. Our base results on EPIE formal show a large increase in metrics over the results proposed (Gamage et al., 2022). We see an increase of 1.24% in precision, 19.6% in recall and 10.9% in F1-score for the minority idiomatic class.

5.5.2. With Re-translation based Loss

Using re-translation based loss improves precision, recall, and F1 scores over binary cross entropy loss on all the datasets. It leads to large gains on `theidioms`, `theidioms 1-1`, `formal`, `gtrans`, and `gpt&gtrans`. This can be explained by the fact that these datasets are characterized by more comprehensive and meaningful sentences compared to `MAGPIE` and `VNC`, which often contain phrases and incomplete sentences. We also observe that the translation-based loss exhibits the highest performance on our in-house dataset, `gtrans`, and this outcome is anticipated, as the expressions included in the dataset primarily rely on the translation model's capacity to grasp the genuine meaning of the idiom in its context and substitute it with a literal phrase conveying the same intended meaning. For the `formal` corpus, we see further increases of 3.3% in precision, 3.11% in recall and 3.22% in F1-score over our regular loss model. This clearly shows the superiority of translation-based loss function.

5.5.3. With Cohesion based Loss

We conducted an initial study to use cohesion based score to classify sentences into containing an idiom or not. It showed results of around 70% accuracy and varied according to the quality of the datasets. Incorporating it as an objective during training improved the accuracy further on all the datasets compared to regular

		Precision			Recall			F1			
Dataset	**Method**	**Precision**	**Precision Macro Avg**	**Precision Weighted Avg**	**Recall**	**Recall Macro Avg**	**Recall Weighted Avg**	**F1**	**F1 Macro Average**	**F1 Weighted Average**	**Accuracy**
MAGPIE	Regular Cross Entropy Loss	[94.1,99.27]	96.68	98.74	[93.64,99.32]	96.48	98.74	[93.87,99.3]	96.58	98.74	98.74
	Translation Retranslation Loss	[93.96,**99.31**]	96.64	98.76	[**93.99**,99.31]	**96.65**	98.76	[93.98,99.31]	96.64	98.76	98.76
	Cohesion based Loss	[94.22,99.28]	96.75	98.76	[93.77,99.34]	96.55	98.76	[93.99,99.31]	96.65	98.76	98.76
	Combination	[**94.5**,99.29]	**96.89**	**98.79**	[93.78,**99.37**]	96.58	**98.8**	[**94.14**,**99.33**]	**96.73**	**98.79**	**98.8**
VNC	Regular Cross Entropy Loss	[97.19,99.64]	98.41	99.43	[96.14,99.74]	97.94	99.43	[96.66,99.69]	98.17	99.43	99.43
	Translation Retranslation Loss	[97.99,**99.81**]	98.9	99.66	[**97.99**,99.81]	98.9	99.66	[97.99,99.81]	98.9	99.66	99.66
	Cohesion based Loss	[98.13,99.76]	98.94	99.62	[97.37,99.83]	98.6	99.62	[97.75,99.79]	98.77	99.62	99.62
	Combination	[**98.45**,**99.81**]	**99.13**	**99.7**	[**97.99**,**99.86**]	**98.92**	**99.7**	[**98.22**,**99.83**]	**99.03**	**99.7**	**99.7**
theidioms	Regular Cross Entropy Loss	[86.61,95.33]	92.07	95.75	[87.37,97.37]	92.36	95.73	[86.98,97.45]	92.21	95.74	95.73
	Translation Retranslation Loss	[91.60,98.68]	95.13	97.52	[93.24,98.33]	95.78	97.5	[92.40,98.50]	95.45	97.51	97.5
	Cohesion based Loss	[91.62,**98.83**]	95.22	**97.65**	[**94.03**,98.32]	**96.17**	**97.62**	**[92.8,98.57]**	**95.69**	**97.63**	**97.62**
	Combination	[**91.76**,98.77]	**95.26**	97.63	[93.73,**98.36**]	96.05	97.61	[92.73,98.56]	95.65	97.61	97.61
formal	Regular Cross Entropy Loss	[90.04,99.18]	94.6	97.89	[95.02,98.29]	96.65	97.82	[92.46,98.73]	95.59	97.84	97.83
	Translation Retranslation Loss	[93.34,99.69]	96.52	98.8	[98.13,98.86]	98.49	98.76	[95.68,99.27]	97.48	98.77	98.75
	Cohesion based Loss	[92.47,**99.75**]	96.11	98.73	[**98.51**,98.69]	**98.6**	98.67	[95.39,99.22]	97.31	98.68	98.67
	Combination	[**93.71**,99.70]	**96.71**	**98.87**	[98.22,**98.92**]	98.57	**98.82**	[**95.92**,**99.31**]	**97.61**	**98.84**	**98.83**
gtrans	Regular Cross Entropy Loss	[85.93,93.87]	89.9	92.38	[72.39,97.27]	84.83	92.61	[78.54,95.53]	87.04	92.36	92.61
	Translation Retranslation Loss	[**86.94**,**96.71**]	**91.83**	**94.89**	[85.68,**97.03**]	**91.36**	**94.91**	[**86.30**,**96.87**]	**91.59**	**94.9**	**94.91**
	Cohesion based Loss	[86.76,**96.71**]	91.74	94.85	[**85.69**,96.99]	91.33	94.87	[86.21,96.85]	91.53	94.86	94.87
	Combination	[86.86,96.58]	91.72	94.76	[85.07,**97.03**]	91.05	94.79	[85.94,96.80]	91.38	94.77	94.79
gpt&gtrans	Regular Cross Entropy Loss	[80.4,97.84]	89.12	96.09	[80.79,97.78]	89.29	96.06	[80.53,97.81]	89.17	96.07	96.07
	Translation Retranslation Loss	[83.91,98.85]	**91.38**	97.34	[89.83,98.06]	93.94	97.23	[86.74,98.45]	92.59	97.27	97.23
	Cohesion based Loss	[83.05,**99.02**]	91.03	**97.41**	[**91.37**,97.91]	**94.62**	**97.25**	[**86.99**,**98.46**]	**92.73**	**97.3**	**97.25**
	Combination	[**83.97**,98.83]	91.4	97.33	[89.64,**98.08**]	93.86	97.23	[86.70,98.45]	92.58	97.26	97.22
theidioms 1-1	Regular Cross Entropy Loss	[66.53,92.80]	79.67	88.91	[57.58,94.97]	76.27	89.44	[61.73,93.87]	77.8	89.12	89.44
	Translation Retranslation Loss	[72.47,93.24]	82.85	90.17	[59.90,**96.05**]	77.97	90.7	[65.58,94.63]	80.1	90.33	90.56
	Cohesion based Loss	[71.88,**93.49**]	82.69	90.3	[**61.59**,95.82]	**78.71**	90.75	[**66.37**,94.64]	80.18	90.45	90.75
	Combination	[**72.84**,93.40]	**83.11**	**90.36**	[60.89,**96.05**]	78.47	**90.85**	[66.31,**94.71**]	**80.51**	**90.51**	**90.85**

Table 2: Results of applying idiom-based custom loss function on several datasets

binary cross entropy loss. As observed for translation-based loss, it leads to large gains on the `theidioms`, `theidioms 1-1`, `formal`, `gtrans`, and `gpt&gtrans`, and performs the best on the `theidioms` and `gpt&gtrans` datasets because these datasets contain sentences which are more complete than `MAGPIE` and `VNC`. For `formal` corpus, we see further increases of 2.43% in precision, 3.49% in recall and 2.93% in F1-score over our regular loss model. This observation aligns perfectly with the fundamental concept of our metric. It underscores that idioms embedded within highly cohesive sentences are more readily identifiable as being idiomatic usages of those phrases.

5.5.4. With combination of losses

Using a combination of both losses improves the accuracy values on `MAGPIE`, `VNC`, `formal`, and `theidioms 1-1` and is very close to the accuracies of translation-based or cohesion-based loss functions for other datasets. In `formal` corpus, we observe notable improvements: precision increases by 3.67%, recall by 3.2%, and F1-score by 3.46% compared to our regular loss model. These discoveries validate the efficacy of utilizing both semantic cohesion and dissimilarity of idiomatic phrases within their contextual environments for our task. Instances penalized by both metrics typically represent confidently idiomatic expressions, which the model should strive to accurately classify.

5.5.5. Cross-domain performance across datasets

We trained our models on one dataset and tested them on another to measure the generalization ca-

		Precision			Recall			F1			
Train, Test	**Method**	**Precision**	**Precision Macro Avg**	**Precision Weighted Avg**	**Recall**	**Recall Macro Avg**	**Recall Weighted Avg**	**F1**	**F1 Macro Average**	**F1 Weighted Average**	**Accuracy**
theidioms, gtrans	Regular Cross Entropy Loss	[84.73,96.41]	90.39	94.11	[84.78,96.29]	90.54	94.1	[84.57,96.35]	90.46	94.11	94.1
	Translation Retranslation Loss	[**89.3**,98.21]	93.76	96.51	[92.46,**97.39**]	94.93	96.45	[90.85,97.8]	94.33	96.48	96.45
	Cohesion based Loss	[**89.3**,**98.39**]	**93.84**	**96.65**	[**93.21**,97.37]	**95.29**	**96.57**	[**91.21**,**97.87**]	**94.54**	**96.6**	**96.57**
	Combination	[89.20,97.97]	93.59	96.3	[91.42,**97.39**]	94.4	96.25	[90.28,97.68]	93.98	96.27	96.25

Table 3: Results showing transfer capabilities of our models. The model is trained on theidioms and tested on gtrans.

pabilities of the model and how our methodology may improve this capability. We trained the model on the theidioms dataset and tested on gtrans dataset. Table 3 shows the result. Our custom loss function based approach showcases impressive transfer capabilities.

5.5.6. Comparison with DISC

We compared our models with DISC (Zeng and Bhat, 2021), a state-of-the-art approach for idiom token classification. We refer to the accuracy values reported in the paper to compare our technique with theirs. We kept the same train-test split for MAGPIE and VNC dataset. It should also be noted that DISC was trained for 600 epochs while our models were trained for only 5 epochs. Table 4 compares the **sequence accuracies** of DISC and our model. Sequence accuracy is considered as a better metric to capture the performance of such models (Zeng and Bhat, 2021). It is clear that our model outperforms DISC in sequence accuracy. This can be explained by our model's capabilities in distinguishing between the literal and figurative idiomatic usages, possible through custom loss function training.

Dataset	**Method**	**Sequence Accuracy**
MAGPIE	Regular Cross Entropy Loss	90.19
	Translation Retranslation Loss	91.31
	Cohesion based Loss	91.46
	Combination	**91.51**
	DISC[3]	87.47
VNC	Regular Cross Entropy Loss	93.75
	Translation Retranslation Loss	**96.88**
	Cohesion based Loss	**96.88**
	Combination	**96.88**
	DISC	93.31

Table 4: Comparing DISC, a state-of-the-art idiom detection model with our technique on 2 datasets

6. Discussion

When we consider the examples where the DISC approach is making incorrect predictions, for instance - "Dragons can lie for dark centuries brooding over their treasures, bedding down on frozen flames that will never see the light of day." The DISC approach incorrectly predicts only a portion of the complete expression - "see the light of day" as idiomatic, whereas our model correctly identifies the entire expression. Similarly for - "Given a method, we can avoid mistaken ideas which, confirmed by the authority of the past, have taken deep root, like weeds in men's minds." where the DISC model predicts "weeds in men's minds" as the idiomatic expression with the correct instance being "taken deep root". Our models do not falter in this case and predict all tokens for this example correctly.

In instances where the cohesion-based approach outperforms combined approaches, it is noteworthy that the Multi-Word Expressions (MWEs) are not consistently translated as expected. Consequently, the incorporation of the translation score tends to diminish overall performance. On the other hand, the translation-only model demonstrates an ability to enhance results compared to the baseline, as it successfully captures anticipated translations for certain expressions, contributing to improved overall performance.

We manually analyze the different errors that our models make on the VNC and EPIE formal datasets to gain insights into the idiom identification abilities and shortcomings in Table 5. We have categorized the errors into 5 major cases and we present examples of each type. Case 1 is where the correct idiomatic expression is identified fully but an alternate expression has also been tagged as idiomatic. This can be thought of as a limitation of the datasets rather than that of our models, as our datasets label at most one expression as idiomatic in each sentence. The second case is where an alternate expression is labeled. The reasoning for this is similar to the previous case as there may be multiple expressions that could possibly be idiomatic and our model is identifying one of them. In the third case, our model correctly identifies the idiom but also tags words surrounding the idiom. This can be ascribed to the alterna-

Error Type	Sentence with PIE	Prediction
Multiple Expressions Predicted	I then walked across to the photographers and lost my temper and then *lost my head*.	lost my temper , lost my head
Alternate expression detected	Cantona will have to *kick his heels* on the sidelines if the manager had his way.	had his way
Extra tokens surrounding expression	Julia had her *attention caught* by the commotion.	attention caught by
Partial	His blistering turn of speed and attitude *made him an instant hit* with the fans.	hit
Predicting Nothing	Everyone talks about *hitting a wall* at the 24 mile mark.	Empty String

Table 5: Different error types along with examples and the incorrect prediction. The ground truth values have been colored blue in sentences.

tive labeling of the identical expression in different occurrences. The fourth case "Partial", constitutes instances where only a segment of the idiomatic expression is identified, with the specific localization of the entire idiom boundary remaining imprecise. The last error category involves the absence of predictions when the model fails to recognize idiomatic usage, even when it is present. The effectiveness of our model is contingent upon the caliber of annotation and various other external factors.

7. Future Work

The latest advancements in Natural Language Processing (NLP) have led to the extensive utilization of a range of transformer-based models. We can adjust our own loss functions to refine different architectures effectively. We can create an intuitive and efficient tool utilizing these fine-tuned models to detect an idiom in a given sentence. This tool should offer a straightforward and accessible experience for a broad range of users, with minimal technical expertise required. To continuously improve the overall performance of our models, we can systematically address each identified error category. This might involve analyzing error patterns and refining the fine-tuning process accordingly.

8. Acknowledgements

We are grateful to the anonymous reviewers for their insightful comments which substantially improved this manuscript. This work was performed using *Sharanga*, the high performance computing cluster at the BITS Pilani Hyderabad Campus.

9. Bibliographic References

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. *Handbook of natural language processing*, 2:267–292.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Rhys Biddle, Aditya Joshi, Shaowu Liu, Cecile Paris, and Guandong Xu. 2020. Leveraging sentiment distributions to distinguish figurative from literal health reports on twitter. In *Proceedings of The Web Conference 2020*, WWW '20, page 1217–1227, New York, NY, USA. Association for Computing Machinery.

J Briskilal and CN Subalalitha. 2022. Classification of Idiomatic Sentences Using AWD-LSTM. In *Expert Clouds and Applications: Proceedings of ICOECA 2021*, pages 113–124. Springer.

BNC Consortium. 2007. British National Corpus, XML edition. Oxford Text Archive.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The VNC-tokens dataset. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 19–22. Citeseer.

Silvio Cordeiro, Carlos Ramisch, Marco Idiart, and Aline Villavicencio. 2016. Predicting the compositionality of nominal compounds: Giving word embeddings a hard time. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1986–1997. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2018. Examining the tip of the iceberg: A data set for idiom translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Afsaneh Fazly and Suzanne Stevenson. 2006. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 337–344.

Gihan Gamage, Daswin De Silva, Achini Adikari, and Damminda Alahakoon. 2022. A BERT-based Idiom Detection Model. In *2022 15th International Conference on Human System Interaction (HSI)*, pages 1–5.

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 279–287, Marseille, France. European Language Resources Association.

Callum Hughes, Maxim Filimonov, Alison Wray, and Irena Spasić. 2021. Leaving no stone unturned: flexible retrieval of idiomatic expressions from a large text corpus. *Machine Learning and Knowledge Extraction*, 3(1):263–283.

Sami Itkonen, Jörg Tiedemann, and Mathias Creutz. 2022. Helsinki-NLP at SemEval-2022 Task 2: A Feature-Based Approach to Multilingual Idiomaticity Detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 122–134.

Changsheng Liu and Rebecca Hwa. 2017. Representations of context in recognizing the figurative and literal usages of idioms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3230–3236. AAAI Press.

Changsheng Liu and Rebecca Hwa. 2019. A generalized idiom usage recognition model based on semantic compatibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 01, pages 6738–6745.

Pengfei Liu, Kaiyu Qian, Xipeng Qiu, and Xuanjing Huang. 2017. Idiom-aware compositional distributed semantics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1213, Copenhagen, Denmark. Association for Computational Linguistics.

Vasudevan Nedumpozhimana, Filip Klubička, and John D Kelleher. 2022. Shapley idioms: Analysing BERT sentence embeddings for general idiom token identification. *Frontiers in Artificial Intelligence*, 5:813967.

Jing Peng and Anna Feldman. 2017. Automatic idiom recognition with word embeddings. In *Information Management and Big Data: Second Annual International Symposium, SIMBig 2015, Cusco, Peru, September 2-4, 2015, and Third Annual International Symposium, SIMBig 2016, Cusco, Peru, September 1-3, 2016, Revised Selected Papers 2*, pages 17–29. Springer.

Jing Peng, Anna Feldman, and Ekaterina Vylomova. 2014. "Classifying Idiomatic and Literal Expressions Using Topic Models and Intensity of Emotions". In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2019–2027, Doha, Qatar. Association for Computational Linguistics.

Priyanka and R.M.K. Sinha. 2014. A system for identification of idioms in hindi. In *2014 Seventh International Conference on Contemporary Computing (IC3)*, pages 467–472.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of 5th international joint conference on natural language processing*, pages 210–218.

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.

Prateek Saxena and Soma Paul. 2020. EPIE Dataset: A Corpus For Possible Idiomatic Expressions.

Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1002–1010.

Patrizia Tabossi, Rachele Fanari, and Kinou Wolf. 2008. Processing idiomatic expressions: Effects of semantic compositionality. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 34(2):313.

Patrizia Tabossi, Rachele Fanari, and Kinou Wolf. 2009. Why are idioms recognized fast? *Memory & cognition*, 37:529–540.

Minghuan Tan and Jing Jiang. 2021. Does BERT Understand Idioms? A Probing-Based Empirical Study of BERT Encodings of Idioms. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1397–1407, Held Online. INCOMA Ltd.

Simone Tedeschi, Federico Martelli, and Roberto Navigli. 2022. Id10m: Idiom identification in 10 languages. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2715–2726.

Rakesh Verma and Vasanthi Vuppuluri. 2015. A new approach for idiom identification using meanings and the web. In *Proceedings of the international conference recent advances in natural language processing*, pages 681–687.

Ziheng Zeng and Suma Bhat. 2021. Idiomatic expression identification using semantic compatibility. *Transactions of the Association for Computational Linguistics*, 9:1546–1562.

Ziheng Zeng and Suma Bhat. 2022. Getting BART to Ride the Idiomatic Train: Learning to Represent Idiomatic Expressions. *Transactions of the Association for Computational Linguistics*, 10:1120–1137.

Ziheng Zeng, Kellen Tan Cheng, Srihari Venkat Nanniyur, Jianing Zhou, and Suma Bhat. 2023. IEKG: A Commonsense Knowledge Graph for Idiomatic Expressions. *arXiv preprint arXiv:2312.06053*.