Information Fusion 86–87 (2022) 30–43

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

MetaPro: A computational metaphor processing model for text pre-processing

Rui Mao [a], Xiao Li [b], Mengshi Ge [a], Erik Cambria [a,*]

[a] Nanyang Technological University, Singapore
[b] University of Reading, UK

ARTICLE INFO

MSC:
68T50

Keywords:
Metaphor identification
Metaphor interpretation

ABSTRACT

Metaphor is a special linguistic phenomenon, challenging diverse natural language processing tasks. Previous works focused on either metaphor identification or domain-specific metaphor interpretation, e.g., interpreting metaphors with a specific part-of-speech, metaphors in a specific application scenario or metaphors with specific concepts. These methods cannot be used directly in everyday texts. In this paper, we propose a metaphor processing model, termed MetaPro, which integrates metaphor identification and interpretation modules for text pre-processing. To the best of our knowledge, this is the first end-to-end metaphor processing approach in the present field. MetaPro can identify metaphors in a sentence on token-level, paraphrasing the identified metaphors into their literal counterparts, and explaining metaphoric multi-word expressions. It achieves state-of-the-art performance in the evaluation of sub-tasks. Besides, the model can be used as a text pre-processing method to support downstream tasks. We examine the utility of MetaPro text pre-processing on a news headline sentiment analysis task. The experimental results show that the performance of sentiment analysis classifiers can be improved with the pre-processed texts.

1. Introduction

Metaphor is defined as using one or several words to describe a concept that is different from the conventional meaning of the words [1]. Given a sentence, "the comedian *convulsed*[1] the children", "*convulsed*" means that the comedian made the children laugh loudly as if they are convulsive, whereas the children are not literally convulsive. Thus, "*convulsed*" is a metaphor in the context. We do not explicitly distinguish metaphors from other figurative languages, such as simile, metonymy, personification, and idiom, which is in line with the metaphor definition in many widely recognized corpora [2–4].

Since metaphors do not take the conventional meanings, these expressions are particularly challenging for natural language processing (NLP). For example, given a sentiment analysis classifier[2] from AllenNLP, the classifier identifies the example sentence, "the comedian *convulsed* the children" as negative, incorrectly (see Fig. 1a). However, if the sentence is paraphrased as "the comedian amused the children", the classifier can yield a positive prediction, correctly (see Fig. 1b). The paraphrasing of metaphors is supportive for several NLP tasks including machine translation [5,6], sentiment analysis [7], question answering [8] and intention mining [9]. Such observations motivate us to study an end-to-end metaphor processing method for improving downstream NLP tasks.

Linguistic metaphor processing consists of two independent tasks, namely metaphor identification and interpretation [10]. The research in metaphor identification was more popular than that of metaphor interpretation, because of the absence of large annotated datasets and learning corpora for metaphor interpretation. Metaphor identification was widely studied with supervised sequence tagging learning [11–16] or dependent word pair classifications [17,18]. Previous metaphor interpretation studies were domain-specific, focusing on a specific part-of-speech (PoS) [5,6,19–22], a specific concept domain [23] or a specific application scenario [24,25]. In this paper, we propose a metaphor processing model — MetaPro[3], combining metaphor identification and interpretation modules. MetaPro can identify metaphors

* Corresponding author.
E-mail addresses: mao.r@ruimao.tech (R. Mao), li.x@ruimao.tech (X. Li), mengshi001@e.ntu.edu.sg (M. Ge), cambria@ntu.edu.sg (E. Cambria).

[1] Italics denote metaphors in this paper.

[2] https://demo.allennlp.org/sentiment-analysis/roberta-sentiment-analysis Accessed 18 March 2021.

[3] MetaPro is deployed at https://metapro.ruimao.tech

[4] A single-word metaphor means that an individual word constitutes a metaphoric expression in a sentence. The whole spectrum of metaphoric expressions covers single-word metaphor expressions and metaphoric MWEs (see the metaphor definition above), where a single-word metaphor can be interpreted individually, while a metaphoric MWE has to be interpreted as a whole.

https://doi.org/10.1016/j.inffus.2022.06.002
Received 21 August 2021; Received in revised form 25 December 2021; Accepted 8 June 2022
Available online 26 June 2022
1566-2535/© 2022 Elsevier B.V. All rights reserved.

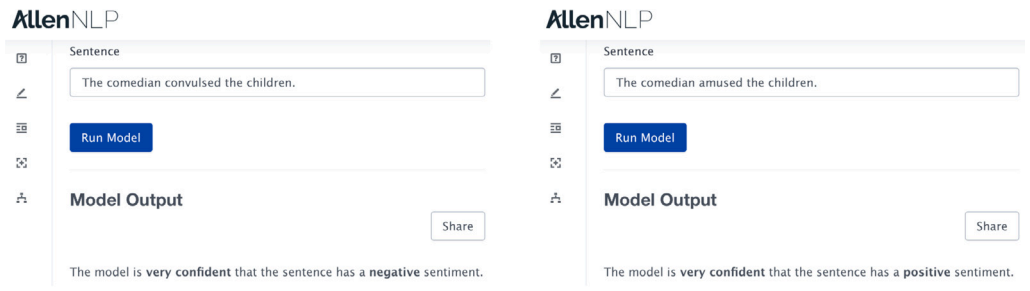

Fig. 1. Sentiment analysis for (a) a metaphoric input, "the comedian convulsed the children"; and (b) its literal counterpart, "the comedian amused the children".

in a sentence on token-level, then paraphrase single-word metaphors or explain metaphoric multi-word expressions (MWEs)[4]. To the best of our knowledge, this is the first end-to-end metaphor processing approach. Thus, MetaPro can be used as a text pre-processing method.

We mean to develop a practical model for supporting NLP techniques, rather than addressing all linguistic metaphor issues in this work. Thus, we focus on identifying and interpreting metaphoric open-class words, say verbs, nouns, adjectives, and adverbs[5] in single word metaphors and idiomatic MWEs, and the improvement of downstream NLP techniques. We believe that interpreting metaphors in these PoS can better support downstream tasks in semantics-related NLP practices, because closed-class words have little lexical meanings [28], e.g., given "you are *in* danger", Lakoff [2] argued that "*in*" is metaphoric, because it is conceptually understood as BEING IN A HARMFUL LOCATION in this context, where the source concept, A HARMFUL LOCATION is different from the target A HARMFUL SITUATION. For sentiment analysis, e.g., "danger" has more semantic information than "*in*", because "*in*" mainly expresses the grammatical relationship between "you" and "danger" in the sentence as a function word. Besides, removing "*in*" does not change the sentiment polarity prediction in the AllenNLP sentiment analysis classifier, which can be explained by the finding of the input reduction of Feng et al. [29]. Thus, the interpretation of metaphoric open-class words is more important than interpreting metaphors in other PoS in practice. For interpreting metaphoric MWEs, we focus on explaining idiomatic MWEs. This is due to the fact that idioms make up a significant part of metaphoric MWEs [3,30]. Besides, interpreting other types of metaphoric expressions with more than one word, such as extended metaphor (the interpretation of metaphoric MWEs is based on a discourse-level) and metaphorical inference (the interpretation is based on the conceptual mappings of source and target domains of MWEs) is limited by current techniques, learning resources and theoretical foundations in both linguistic and computational linguistic communities [10]. As a result, we mean to support downstream NLP techniques practically. The performance of our method is ultimately evaluated in a downstream task.

We examine our model, MetaPro on two metaphor identification tasks, a metaphor interpretation task, and a sentiment analysis task. For metaphor identification, we examine MetaPro on the largest all-word annotated metaphor dataset [3]. MetaPro outperforms the strongest baselines by 1.3% F1 scores on average with respect to open-class metaphor identification and all-PoS metaphor identification tasks. For metaphor interpretation, MetaPro exceeds the baselines across all three evaluation dimensions (coherence, semantic completeness, and literality), according to human evaluation results. By using MetaPro as a text pre-processing method, the average gain of three sentiment analysis APIs from NLTK [31], AllenNLP[6] and Microsoft Azure Text Analytics[7] achieves 4.0% F1 in SemEval2017 Task 5 news headline dataset [32]. We also observe an average gain of 1.9% F1 in a strong news headline sentiment analysis task-specific classifier.

The contribution of this work is threefold: (1) we propose the first end-to-end metaphor processing model, termed MetaPro for the text pre-processing; (2) The metaphor identification and interpretation modules of MetaPro achieve state-of-the-art performance on each task; (3) We demonstrate that MetaPro can support sentiment analysis classifiers in classifying metaphoric texts.

2. Related works

Metaphor is a special linguistic phenomenon, which has been widely studied in the communities of linguistics and computational linguistics.

Metaphor studies in linguistics have proposed several methods for identifying metaphors [33–39]. Pragglejaz [38] believed that there is a semantic contrast between the contextual meaning and the basic meaning of a metaphor in their Metaphor Identification Procedure (MIP). Thus, one can identify a metaphor by interpreting the basic meaning and the context meaning, and analyzing their semantic contrast. Wilks [33][34] proposed a Selectional Preference Violation (SPV) theory for identifying metaphors. A metaphor violates the selectional preference of its context, where the selectional preference can be measured by a word co-occurrence within a certain semantic category of contextual words. Lakoff and Johnson [36] argued that metaphor is not only a linguistic phenomenon, but also reflects human thoughts and behaviors in their Conceptual Metaphor Theory (CMT). They believed that metaphor is a conceptual projection from source to target domains. Thus, it can be identified by mapping concepts from two different domains.

Computational metaphor processing can be categorized as linguistic metaphor processing and conceptual metaphor processing. The former studies the surface realization of metaphors [10], e.g., identifying metaphors and interpreting the metaphors from semantic aspects, while the later investigates metaphor concept mapping mechanisms between source and target domains [40]. Since we mean to support downstream NLP tasks, we focus on linguistic metaphor processing in this work. There are two popular tasks in linguistic metaphor processing, namely metaphor identification and interpretation.

Metaphor identification is the most widely studied sub-task in computational metaphor processing [41,42]. Previous works identified metaphoricity on sentence-level [43,44], word-pair-level [17,18], or token-level [5,45] with different feature engineering methods, e.g., word embeddings, topic modeling, image features, dependency-tree-based features, lexical resources, and word co-occurrences. Compared with sentence-level and phrase-level approaches, token-level

[5] Linguistic studies [26,27] also categorized proper nouns and interjections as open-class words. We identify these metaphors without interpreting them in this paper, because interpreting these types of metaphors, e.g., "Steve Jobs is the *Michael Jackson* of the tech world" and "*well*, very worried" requires very specific domain knowledge. We will study the interpretation of these metaphors in future work.

[6] https://allennlp.org Accessed 18 March 2021.

[7] https://azure.microsoft.com Accessed 18 March 2021.

approaches can identify the exact metaphorical words in a full sentence. Mao et al. [5] proposed an unsupervised verbal metaphor identification method by modeling the semantic similarity between a metaphorical verb and its literal counterpart. Song et al. [46] also focused on verbal metaphor identification by explicitly modeling the grammatical, sentential and semantic relationship between a verb and its context. However, the implicit condition of previous token-level methods is that the position of a target word whose metaphoricity is to be identified has been given in advance, which is not highlighted in real-word texts. Currently, deep neural network (DNN)-based sequence tagging methods are widely applied in metaphor identification [11–16], because they can identify metaphors on token-level for all words in a sentence. Hence, the sequence tagging metaphor identification methods are more readily applied for metaphor interpretation. MIP and SPV were incorporated as linguistic features to improve sequential metaphor identification performance [13,47]. Semantic and syntactic features, e.g., word clusters and PoS tags were also widely used features in DNN-based models [11,14,48].

Metaphor interpretation targets to interpret the real meaning of a metaphor, e.g., paraphrasing metaphors into their literal counterparts in linguistic metaphor processing. Many of previous works focused on a specific application scenario, e.g., answering metaphorical queries of users about Unix [24] and analyzing mental state descriptions [25]; a specific concept domain, e.g., motion verbs [23]; specific dependency relationships, e.g., adjective-noun and subjective-verb-object relations [49]; or a specific class of PoS, e.g., verbal metaphor interpretation [5,19,20] and nominal metaphor interpretation [21,22], based on manually defined knowledge, lexical resources, web searching or word co-occurrence modeling. However, one cannot directly use these metaphor interpretation approaches for interpreting everyday texts from different domains. Recently, Wan et al. [50] used gloss as additional features for metaphor identification and interpretation. However, the output of such a metaphor interpretation method that identifies the gloss of a metaphor cannot be processed by a downstream task.

In contrast to previous metaphor identification and interpretation works, we combine the two tasks as a unified metaphor processing task, processing texts that are not limited to specific conceptual or practical domains. Both input and output of our model are natural language. Thus, our method can be used as a text pre-processing technique to support downstream NLP tasks and linguistic learners.

3. Methodology

As seen in Fig. 2, there are two technical modules in MetaPro, namely metaphor identification (the blue) and metaphor interpretation (the green). The metaphor identification module means to detect metaphors on token-level from an input sequence. The metaphor interpretation module means to interpret the identified metaphors by paraphrasing single-word metaphors into their literal counterparts, or pairing identified metaphoric MWEs with their dictionary meanings. Finally, the output of MetaPro is given by integrating (the gray box) the paraphrases of single-word metaphors and the paired meanings of metaphoric MWEs into the literal context of the input sequence. In the output sequence, the single-word metaphors are replaced with their paraphrases (the literal counterparts). The metaphoric MWEs are explained with a clause beginning with "where", e.g., given "the comedian *convulsed* the children in a *red letter* day", the metaphor identification module detects "*convulsed*", "*red*" and "*letter*" are metaphors. The interpretation module paraphrases "convulsed" (a single-word metaphor) as "amused", explaining "red letter day" (a metaphoric MWE) as "a day of significance", because an MWE classifier detects that "*red letter* day" is an MWE. Finally, the output is integrated as "the comedian amused the children in a red letter day, where 'red letter day' means that a day of significance". The technical details of the metaphor identification and interpretation modules are introduced in the following subsections.

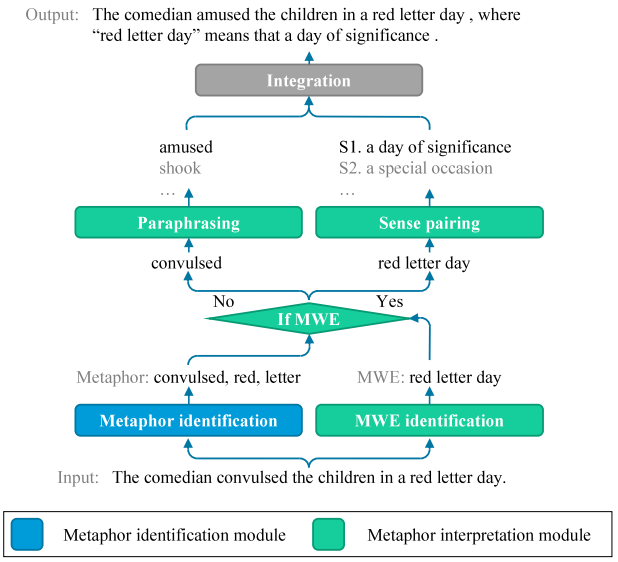

Fig. 2. The overall framework of MetaPro. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1. Metaphor identification

We adopt a multi-task learning (MTL) model for jointly learning metaphor identification and PoS tagging[8]. The model and the soft-parameter sharing method (Gated Bridging Mechanism) were firstly proposed in the work of Mao and Li [16]. The motivation is: (1) MTL reduces the risk of overfitting, learning richer features from different tasks [52]; (2) Previous works have demonstrated the utilities of MTL in metaphor identification [15,16,53]; (3) PoS tags are sufficient features for sequential metaphor identification [11,14,48]. Fig. 3 shows (a) the framework of our MTL model with two sub-tasks, (b) the structure of Gated Bridging Mechanism. The equations below (Eqs. (1)–(11)) are cited from Mao and Li [16].

Given an input sequence $t_1, \ldots, t_L$ (t is a token; L denotes the length), RoBERTa [54] is employed as an encoder, encoding the input sequence, yielding sharing hidden states H^s in MTL.

$$H^s = RoBERTa(t_1, \ldots, t_L). \tag{1}$$

Upon the sharing encoder, we use Transformers [55] to construct task-specific towers. Previous works showed that sharing parameters between task-specific towers can further improve model performance [52]. Thus, we introduce a Gated Bridging Mechanism $GBM_{\phi_{i,j}}(\cdot)$ for soft-parameter sharing, where i denotes Block i in a task-specific tower, j is a tower that learns Task j (τ_j). ϕ denotes learned parameters in a function. A block consists of a Gated Bridging Mechanism and a Transformer layer (see Fig. 3a). The output of a Gated Bridging Mechanism is given by taking the Transformer hidden states from a previous bock ($i-1$) across all towers

$$G_i^{\tau_j} = GBM_{\phi_{i,j}}(H_{i-1}^{\tau_1}, \ldots, H_{i-1}^{\tau_j}, \ldots). \tag{2}$$

The details of $GBM_{\phi_{i,j}}(\cdot)$ will be explained later.

Next, the hidden states of Transformer in each block are given by

$$\begin{cases} H_0^{\tau_j} = Trans_{\phi_{0,j}}(H^s), \\ H_i^{\tau_j} = Trans_{\phi_{i,j}}(G_i^{\tau j}), \quad 0 < i \le n. \end{cases} \tag{3}$$

For Block 0, Transformer ($Trans_{\phi_{0,j}}(\cdot)$) takes the sharing hidden states (H^s) as input. For other blocks, Transformer ($Trans_{\phi_{i,j}}(\cdot)$) takes the

[8] The Universal Dependency PoS labels are obtained via spaCy toolkit [51].

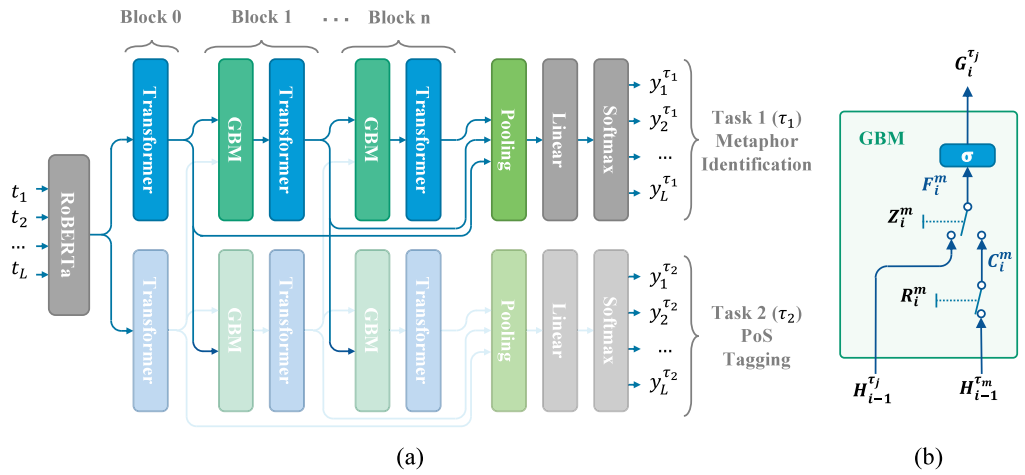

Fig. 3. The framework of metaphor identification module in MetaPro. The figure is adapted from the work of Mao and Li [16]. (a) The multi-task learning model with Gated Bridging Mechanisms (GBM). There are two subtasks (τ_1, τ_2) in the model. Each subtask consists of $n+1$ blocks. t is an input token. L is the length of the input sequence. y^{τ_1} is a metaphoricity label. y^{τ_2} is a PoS tag. (b) The structure of Gated Bridging Mechanism. i denotes Block i. H is Transformer hidden states. τ_j and τ_m are two different subtasks in multi-task learning, where τ_j is the focused task in a private Tower j. m is a neighbor tower, where $m \neq j$. R_i^m is a reset gate. C_i^m is new current states. Z_i^m is an update gate. F_i^m is the fusion of H^{τ_j} and C_i^m. σ is sigmoid activated fully connected layer. $G_i^{\tau_j}$ is the output.

output of a Gated Bridging Mechanism ($G_i^{\tau j}$) in the same block and tower as input.

We fuse all the Transformer hidden states in each tower via a weighted sum pooling strategy, because the findings of Liu et al. [56] and Mao et al. [48] demonstrated that different Transformer layers have different utilities in modeling semantic and syntactic information. The hidden states for the pooling of Tower j ($H_{pool}^{\tau_j}$) is given by

$$H_{pool}^{\tau_j} = \sum_{i=0}^{n} \alpha_i^{\tau_j} H_i^{\tau_j}, \tag{4}$$

where, the weight $\alpha_i^{\tau_j} \in \mathbb{R}$ is a parameter for learning.

The predicted task-specific feature ($\hat{Y}^{\tau_j}$) of the input sequence is given by a fully connected layer (fc)

$$\hat{Y}^{\tau_j} = W_{fc}^{\tau_j} H_{pool}^{\tau_j} + b_{fc}^{\tau_j}, \tag{5}$$

where W and b are parameters for learning.

Finally, we use Cross-entropy Loss that integrates the final softmax function in Fig. 3a, where the loss ($\mathcal{L}$) is given by

$$\mathcal{L} = \sum_{\tau_j} \text{CrossEntropy}(\hat{Y}^{\tau_j}, Y^{\tau_j}). \tag{6}$$

Gated Bridging Mechanism: Inspired by Cho et al. [57], Gated Bridging Mechanism uses gating mechanisms as controllers for the filtering and fusing information between different MTL towers, e.g., a reset gate R_i^m is employed for controlling the information flow of Transformer hidden states in a previous block ($i-1$) from a neighbor tower ($H_{i-1}^{\tau_m}$) passing to a private tower.

$$R_i^m = \sigma(W_{\phi_{R,i,j}}^m H_{i-1}^{\tau_m} + b_{\phi_{R,i,j}}^m), \tag{7}$$

where σ denotes the sigmoid activation function. We use j denotes a private tower that processes τ_j in Fig. 3b. Tower m is a neighbor tower, where $m \neq j$. The filtered information (C_i^m) from a neighbor tower in Block i is given by

$$C_i^m = tanh(W_{\phi_{C,i,j}}^m (R_i^m \odot H_{i-1}^{\tau_m}) + b_{\phi_{C,i,j}}^m), \tag{8}$$

where $\odot$ denotes element-wise product. There is a non-linear projection in Eq. (8), because we believe that hidden states in different towers are from different vector spaces. The non-linear projection function can project the hidden states from the neighbor tower space to the space of hidden states of the private tower.

Next, we introduce an update gate Z_i^m. Z_i^m controls if C_i^m fuses with the hidden states ($H_{i-1}^{\tau_j}$) in the previous block of the private tower.

$$Z_i^m = \sigma(W_{\phi_{Z,i,j}}^m H_{i-1}^{\tau_j} + b_{\phi_{Z,i,j}}^m + V_{\phi_{Z,i,j}}^m C_i^m + d_{\phi_{Z,i,j}}^m), \tag{9}$$

where V and d are parameters for learning. The post-fused feature (F_i^m) of a private tower and a neighbor tower is given by

$$F_i^m = Z_i^m \odot H_{i-1}^{\tau_j} + (1 - Z_i^m) \odot C_i^m. \tag{10}$$

As seen in Eq. (10), there is a trade-off between $H_{i-1}^{\tau_j}$ and C_i^m. Fusing more information from a private tower means that Gated Bridging Mechanism rejects more information from a neighbor tower. This function allows Gated Bridging Mechanism to make the best use of information from different towers in the same block.

Finally, the output ($G_i^{\tau_j}$) of Gated Bridging Mechanism $GBM_{\phi_{i,j}}^{\tau_j}(\cdot)$ in τ_j Block i is given by

$$G_i^{\tau_j} = \sigma(W_{\phi_{G,i,j}}^{\tau_j} F_i^m + b_{\phi_{G,i,j}}^{\tau_j}). \tag{11}$$

3.2. Metaphor interpretation

3.2.1. Linguistic hypothesis

We mean to paraphrase a single-word metaphor into its semantically similar literal counterpart. Our linguistic hypothesis mainly depends on MIP and SPV, because MIP and SPV explain the mechanisms of metaphors semantically (see Section 2). The difference is that MIP was proposed for human metaphor identification. A metaphor can be identified by the contrast between its basic and contextual meanings in MIP. SPV is machine-friendly. The selectional preference is measured by the statistics of word co-occurrences, where a literal word that satisfies the selectional preference of its context also frequently appears in its context. This is in line with corpus studies that metaphors take about a third of sentences in typical corpora [3,58,59]. We believe that MIP and SPV are fundamentally similar, because the basic meaning of a metaphor violates the selectional preference of a context in SPV, while the inferred contextual meaning of the metaphor can satisfy the selectional preference. Thus, the basic meaning of a metaphor contrasts the contextual meaning in MIP. On the other hand, the basic and contextual meanings of a literal are similar, because both can satisfy the selectional preference of the context.

Given MIP and SPV, we hypothesize that a literal counterpart which represents the contextual meaning of a metaphor satisfies the selectional preference of the context, thus, having a high co-occurrence frequency, appearing in a context.

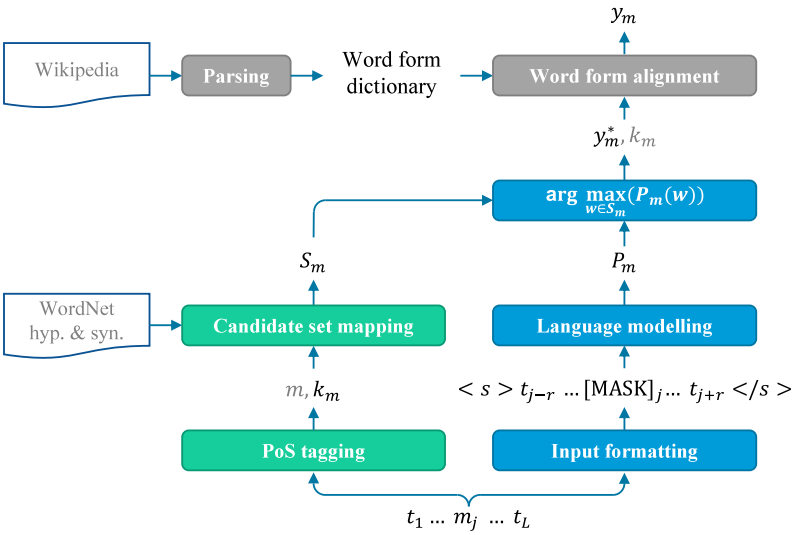

Fig. 4. The paraphrasing workflow in MetaPro. t is a context word. m is an identified metaphor. j is the metaphor position. L is the length of an original input sequence. k_m is the PoS of the metaphor. S_m is a candidate set for paraphrasing m, where $w \in S_m$. r is a window size. P_m is the probability distribution of words appearing in the metaphor position. y_m^* is the best-fit word for the context. y_m is the paraphrase of m, aligning to the word form in PoS k_m. Black texts are inputs and outputs. Gray texts are inputs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2.2. Metaphor paraphrasing

We use a pre-trained Language Model and WordNet [60] hypernyms and synonyms to paraphrase identified single-word metaphors. This method was firstly proposed in the work of Mao et al. [5]. In that work, a Continuous Bag of Words (CBOW)-based word2vec [61] language modeling method was employed for inferring the literal counterpart of a verbal metaphor. However, current deep learning-based Language Models achieve better performance in diverse NLP tasks [54,62,63]. In this work, we use RoBERTa [54] pre-trained Language Model instead of word2vec. We also newly incorporate a word form alignment process to align the word forms of paraphrases to the original metaphor word forms, compared with the work of Mao et al. [5].

RoBERTa consists of multi-Transformer layers. It was trained to predict the probability of a masked word, appearing in the context with a very large corpus (160 GB). Thus, we can use RoBERTa to predict a possible paraphrase that most likely co-occurs with a given context. The works of Mao et al. [5] and Mao et al. [6] demonstrated that hypernyms and synonyms in WordNet are eligible semantic-similar candidates for a word in different sense classes. The hypernyms and synonyms also cover metaphorical sense classes, e.g., "amuse" is a hypernym of "convulse" in the sense of "make someone convulse with laughter" in WordNet. We can use WordNet as a lexical resource to constrain a paraphrase prediction that is semantically similar to the metaphorical sense. The paraphrase is identified as the literal counterpart of a metaphor, according to the hypothesis in Section 3.2.1. The detailed metaphor paraphrasing workflow can be viewed in Fig. 4.

First (the green boxes in Fig. 4), we parse the input sequence with spaCy, obtaining the PoS tags (k) of metaphors. The PoS tags are represented as coarse-grained PoS ($k^{crs} \in \{\text{verb}, \text{noun}, \text{adjective}, \text{adverb}\}$) and fine-grained PoS ($k^{fn}$), respectively. k^{fn} are the Penn Treebank Part of Speech Tags [64], representing both PoS classes and word forms. The hypernyms and synonyms of metaphors are gathered from WordNet. We define a function $C(,)$ that maps an identified metaphor m and its coarse-grained PoS class k_m^{crs} to a candidate set S_m.

$$S_m = C(m, k_m^{crs}). \tag{12}$$

S_m contains the hypernyms and synonyms of m, and their inflections in k_m^{crs}. These words ($w, w \in S_m$) are the paraphrasing candidate words of m.

Next (the blue boxes in Fig. 4), we compute the best substitution word for the metaphor m, based on pre-trained RoBERTa Language Model. We hypothesize that a very large context may bring noise for predicting an optimal paraphrase (see Section 5.2.1). Thus, the full input sequence is pruned, according to a manually defined window size r, where r words before and after the metaphor are included in the Language Model input sequence. A single-word metaphor is replaced with a special token "[MASK]" (see Section 3.2.3 for metaphoric MWE detection). If an original sentence has multiple identified single-word metaphors, we prepare multiple Language Model inputs where each input has a "[MASK]". "<s>" and "</s>" are placed at the beginning and the end of a Language Model input sequence, because these special tokens were defined during the RoBERTa pre-training procedure. Thus, an original sequence $t_1, \ldots, m_j, \ldots, t_L$ is formatted as $\langle s \rangle, t_{j-r}, \ldots, [\text{MASK}]_j, \ldots, t_{j+r}, \langle /s \rangle$. We predict the probability distribution (P_m) of words appearing in the metaphor position with pre-trained RoBERTa Language Model ($LM(\cdot)$).

$$P_m = LM(\langle s \rangle, t_{j-r}, \ldots, [\text{MASK}]_j, \ldots, t_{j+r}, \langle /s \rangle). \tag{13}$$

We find the best-fit word (y_m^*) that has the highest probability co-occurring with the context from the candidate set (S_m) of m. y_m^* is given by

$$\begin{cases} y_m^* = arg \max_{w \in S_m} (P_m(w)), & \text{if } y_m^* \in \{\text{top } u \text{ words}\} \\ y_m^* = m_j, & otherwise. \end{cases} \tag{14}$$

We setup a hyperparameter u, denoting a set of top u words that most likely appear in the context among all RoBERTa predictions (including non-candidates). Given the threshold u, the best-fit word prediction with low confidence (not one of top u predictions) is not used for paraphrasing to avoid ruining the original sentence (see Table 13 for the sensitivity tests of u in Section 5.3).

Finally (the gray boxes in Fig. 4), we align the word form of the best-fit word to that of the original metaphor via a dictionary-based mapping function ($\mathcal{F}(,)$). We develop a dictionary, mapping a lemma and all its possible word forms mutually, based on the Penn Treebank Part of Speech Tags (k^{fn}). The PoS tags, lemmas, mappings and vocabularies in the dictionary are obtained by parsing a Wikipedia dump[9] with spaCy. The mapping function first maps the predicted best-fit word to its lemma form, then mapping the lemma to the form in the

[9] https://dumps.wikimedia.org/enwiki/20170920/ Accessed 1 November 2017.

PoS k_m^{fn} that is same to the original metaphor in the context. Given the fine-grained PoS tag k_m^{fn} of the original metaphor, best-fit word y_m^* and the word form mapping function $\mathcal{F}(,)$, the final paraphrase (y_m) of the metaphor is given by

$$y_m = \mathcal{F}(y_m^*, k_m^{fn}). \tag{15}$$

For example, given $m = convulsed$ in "the comedian *convulsed* the children", where $k_{convulsed}^{fn} = VBD$ (VBD means verb in past tense), the predicted best-fit word at the position of "*convulsed*" is "amuses" ($y_m^* = y_{convulsed}^* = amuses$). Thus, the final paraphrase has to be aligned to the form of "amused" ("amuses" in VBD). Mathematically, $y_m = amused = \mathcal{F}(amuses, VBD)$.

3.2.3. Metaphoric multi-word expression interpretation

Since the interpretation of metaphoric MWEs cannot be paraphrased token-by-token, we employ a simple yet efficient dictionary and rule-based method to identify and interpret metaphoric MWEs. A dictionary and rule-based method is computationally economic and lighter than DNN models. Such a method allows us to identify different metaphoric MWEs on token-level, and match the identified MWEs to the vocabularies of our pre-defined MWE dictionary and the associated meanings from end-to-end. In contrast, to the best of our knowledge, the current sequence tagging-based method [65] and Graph Convolutional Network-based method [66] can only reach the target of identifying the individual tokens that belong to MWEs. As a result, there is still a gap between the identifying of MWE tokens and the pairing of the meanings of MWEs (our target) in these Machine Learning-based methods. Thus, we use the dictionary and rule-based method.

We parse and lemmatize an input sentence with spaCy, first. Then, we pair the sentence features with dependency tripe features and lemma features that are pre-defined in our MWE dictionary via rules to identify MWEs. Finally, a paired MWE is explained by its meaning in the dictionary.

Dependency triple pairing method: Dependency pairing was widely used in searching domains [67–69]. Intuitively, an MWE has a conventional syntactic structure, interacting within the context, because it acts as a united unit, spanning common word boundaries [70]. Thus, we can use a dictionary and rule-based dependency triple (head word lemma, dependency relationship, child word lemma) matching method to detect metaphoric MWEs. Compared with previous dictionary-based MWE detection methods [71,72], our dependency triple pairing method takes advantage of handling gap flexibility, e.g., the method of Ghoneim and Diab [72] is based on a Maximum Forward Matching algorithm. However, they argued that this algorithm cannot handle an MWE with an insertion, e.g., "break up" can be paired with the vocabulary in a dictionary based on Maximum Forward Matching, while "break it up" is an exception. In contrast, target MWEs (MWEs in a sentence) and source MWEs (MWEs in our dictionary) are paired according to the dependency triples in our method. Given "We have been silent for a long while. I want to *break* it *up*", "*break* it *up*" is lemmatized and parsed as ('break', 'prt', 'up') which can successfully match to the pre-defined dependency triple feature of "break up" ('break', 'prt', 'up'), because the head ('break') and child ('up') lemmas have the same dependent relationship ('prt'). The dependency triple pairing rule is:

R1. If the pre-defined dependency triple features of an MWE in the dictionary is a subset of the dependency triple features of a given sentence in lemma forms, the matched head and child words in the sentence are an MWE.

Dependency triple feature preparation: Our metaphoric MWE dictionary covers the vocabulary and meanings of The Idioms[10] (the largest idiom dictionary) and the collection of Agrawal et al. [73] (the largest idiom dataset to the best of our knowledge). We use these two data sources, because they provide sufficient coverage in idioms. Idioms make up our metaphoric MWE dictionary, because idiomatic phrases are figurative [74,75], taking a significant part of metaphoric MWEs [30]. Idioms were also annotated as metaphors in the largest all-word annotated metaphor dataset [3], based on MIP. We remove those idiomatic MWEs from our dictionary, whose meanings can be interpreted by paraphrasing (see Section 3.2.2), e.g., "I don't *buy* (believe) it" and "I *blew* (wasted) it".

The pre-defined dependency triple features in the dictionary are parsed from example sentences and manually selected. We use Google to search example sentences, containing the MWEs (10 sentences per MWE). The example sentences are parsed with spaCy. The principle of selecting our pre-defined dependency triple features of an idiom from the 10 example sentences is that we include open-class words of the idiom and exclude closed-class words as much as possible, because metaphoric open-class words can be identified by the metaphor identification module; Besides, we select those triples that are flexible for fitting different contexts. Thus, not all dependency relationships related to the tokens of an MWE are included as the matching items, e.g., [('poor', 'prep', 'as'), ('mouse', 'compound', 'church')] is the pre-defined dependency triple features for "*as poor as a church mouse*", where ('poor', 'advmod', 'as') is excluded from the list, because the idiom, "*as poor as a church mouse*" is possibly written in the form of "*poor as a church mouse*" in its example sentences, where ('poor', 'advmod', 'as') is not covered by the example sentences. An MWE may have multiple sets of triple features so that they can be better paired with idioms in different syntactic structures, e.g., both [('brew', 'nsubj', 'storm')] and [('brewing', 'compound', 'storm')] can be paired with idiomatic "*storm is brewing*". We do not define a specific list of inclusive dependency relationships, because the selected triple features cannot be precisely governed by specific dependencies. Thus, we manually select the triple features of an idiom, based on the parsing results of the idiom and its 10 different contexts.

Lemma pairing method: We employ an additional lemma pairing rule to support the idiom detection, because not all MWEs can be perfectly represented as dependency triples, e.g., given "we finally decided it was *now or never* to buy the car", the parsed dependency relationships related to the target MWE "*now or never*" are ('be', 'advmod', 'now') and ('buy', 'neg', 'never'), where "be" and "buy" do not belong to the idiomatic MWE. In this case, pairing MWEs by lemma sequences is more effective. Besides, dependency parser performance issues are not uncommon. Thus, lemma pairing is supportive in such a scenario. The lemma pairing rule is:

R2. If a lemmatized string of a sentence contains all lemmatized features of an MWE in the dictionary, the tokens in the sentence are identified as an MWE.

Lemma feature preparation: The lemma features of idioms are obtained from the lemmatized sequences of the collected example sentences (10 for each). We select the chunk of lemma tokens that matches an idiomatic expression in our dictionary as its lemma features. For those MWEs in the dictionary, containing indefinite pronouns, e.g., "*know something inside out*", the MWE is featured as a list of lemma sequence fragments, such as ['know', 'inside out'], where the indefinite pronoun, "something" is excluded.

An additional rule for multi-matching: It is possible that a sentence matches multiple MWEs, e.g., both "*first world*" and "*first world problem*" can match to "they are just the *first word problems* for me" by their lemmas. In this case, the rule is:

R3. If a given sentence matches the features of multiple MWEs, and the feature of an MWE is a subset of the feature of the other matched MWE, the longer one wins.

R3 also works for the dependency triple matching method, e.g., if both "*drop in*" ([('drop', 'prep', 'in')]) and "*a drop in the ocean*" ([('drop', 'prep', 'in'), ('in', 'pobj', 'ocean')]) are matched by a sentence, we will take the later one as an identified MWE in the sentence.

Sense pairing method: Finally, the interpretation of an MWE is given by:

[10] https://www.theidioms.com Accessed 8 February 2021.

R4. If any tokens of an MWE in an input sequence were identified as metaphoric by the metaphor identification module, the MWE is explained with its sense in the dictionary.
R5. If an MWE has multiple senses, we choose the one that is the most semantically similar to the sentence as the sense of the MWE in the context.

The semantic similarity is measured by cosine similarity of the sequence embeddings (v^s) between a sense sequence and the original input sentence. The selected sense ($sense^*$) for output is given by

$$sense^* = \arg \max_{sense \in dict.} cos(v^s_{sense}, v^s_{input}), \tag{16}$$

where v^s is the mean pooling of RoBERTa hidden states of tokens in a sequence.

Totally, 3560 lemma pairing features and 3470 dependency triple pairing features are defined for 3050 unique idiomatic MWEs. Each MWE has 2.7 meanings on average. To the best of our knowledge, this is the largest idiom dictionary that specifies the dependency triple features, lemma features, and the meanings of idioms.

4. Experiments

4.1. Setups

In the metaphor identification module (MetaPro-ID[11]), we employ 2 task-specific towers upon a RoBERTa-large sharing encoder for learning metaphor identification (the main task) and PoS tagging (the auxiliary task). Each task-specific tower consists of 4 Transformer layers and 3 associated Gated Bridging Mechanisms. Each Transformer has 16 heads and 1024 dimensions. We keep the shape of Gated Bridging Mechanism output the same to the Transformer output. Thus, the learned parameters in Gated Bridging Mechanism (W and V in Eqs. (7),(8),(9),(11)) also have 1024 dimensions. We use the default setups of PyTorch [76] for the initialization of learning parameters. The model is trained with a batch size of 4, optimized with Adam optimizer (1e-5 initial learning rate and a Step Decay schedule) [77]. We train the model 20 epochs.[12] The model that has the highest F1 score on a validation set is used for the evaluation of the corresponding testing set. Since RoBERTa-large is used as the sharing encoder, where the input tokens are encoded as Byte-Pairs. In order to keep the length of output metaphoricity labels the same to the length of the input sequence, we use the prediction of the first word-piece token of a word as the prediction for the word.

In the metaphor paraphrasing model, the window size for predicting the probability of a candidate paraphrase is 16 ($r = 16$ in Eq. (13)). We set up a threshold for filtering out unconfident paraphrase predictions whose probabilities are not top 5000 ($u = 5000$ in Eq. (14)) among RoBERTa Language Model predictions.

4.2. Baselines

4.2.1. Metaphor identification

Wu et al. [11] proposed a Convolutional Neural Network (CNN) [78] and Bi-directional Long Short Term Memory (BiLSTM) [79] model for identifying metaphors on token-level. The model uses word2vec, PoS tags, and word2vec clusters as features, achieving the best performance on the 2018 Metaphor Detection Shared Task [41].

Gao et al. [12] proposed a typical sequence tagging model for metaphor identification. The model has a BiLSTM layer and a softmax classifier, using the concatenation of ELMo [80] and GloVe [81] as features.

Mao et al. [13] proposed a linguistic-inspired model by modeling the semantic contrast between a metaphor and its context (SPV). The concatenation of ELMo and GloVe is encoded by BiLSTM first, then the semantic contrast is modeled by using a multi-head contextual attention mechanism.

Dankers et al. [53] proposed MTL models, learning metaphor detection and emotional auxiliary tasks. The best model is given by a BERT-based hard parameter sharing model, where the auxiliary task is predicting valence scores.

Le et al. [15] proposed an MTL model, based on Graph Convolutional Network, learning metaphor identification and Word Sense Disambiguation iteratively. Their model used ELMo, GloVe and index embeddings as features.

Su et al. [14] proposed a reading comprehension paradigm-based model for metaphor identification. The model encodes multi-features, e.g., global and local contexts, question information, and PoS with RoBERTa, yielding state-of-the-art results on the 2020 Metaphor Detection Shared Task [42].

RoBERTa [54] is a state-of-the-art pre-trained Language Model. We use it as a sequence tagging model to evaluate the improvement of our MTL strategy and Gated Bridging Mechanism.

RoBERTa+PoS is introduced as a baseline to compare the performance difference between MTL-based MetaPro-ID and a single task learning-based classifier. RoBERTa+PoS consists of a RoBERTa encoder and 4 Transformer layers upon the RoBERTa encoder. The RoBERTa hidden states are concatenated with one-hot encoded PoS tags, then feeding to the Transformers and a softmax classifier.

4.2.2. Metaphor interpretation

We test two different Language Models, e.g., BERT-large and ALBERT-xxlarge-v2 for metaphor paraphrase evaluation. These Language Models are used in a similar way as our proposed RoBERTa-based method.

BERT [62] is a Transformer-based pre-trained Language Model. It uses context Word Pieces [82] to predict a masked word and the next sentence during the pre-training procedure.

ALBERT [63] is a lighter pre-trained Language Model, compared with BERT. The parameter size is reduced in ALBERT by introducing factorized embedding parameterization and cross-layer parameter sharing methods. Besides, ALBERT computes sentence-order prediction loss instead of the loss of next sentence prediction of BERT in language modeling.

For the metaphoric MWE detection evaluation, we test two Machine Learning models, namely RoBERTa (sequence tagging) and BiLSTM-CRF.

BiLSTM-CRF [83] is a classical sequence tagging model that was used in an idiom detection task by Saxena and Paul [65]. BiLSTM and the Conditional Random Field (CRF) [84] layers allow the model to learn contextual information and the conditional probabilities of tag transition simultaneously. 300-dimension GloVe was employed as input features.

4.2.3. Sentiment analysis

We test three publicly available sentiment analysis APIs from NLTK, AllenNLP, and Microsoft Azure, respectively. These three APIs are selected because of their different features:

Vader [85] is a rule-based method that has been built in the NLTK package. It maps input sequence to a valence-based sentiment lexical dictionary, yielding sentiment scores in positive, negative, and neutral polarities. It achieved outstanding performance in the domains of social media texts, news, movie reviews, and product reviews.

AllenNLP uses a RoBERTa-based deep learning model and Stanford Sentiment Treebank dataset [86] to train a binary sentiment classifier (negative or positive). It achieves 95.11% accuracy on the testing set.

Azure is a commercial NLP toolkit from Microsoft Azure Text Analytics, whereas the details of their sentiment analysis classifier are unknown. We can use the Azure sentiment analysis classifier to obtain

[11] Without further specification, MetaPro-ID is based on a RoBERTa-large encoder. The subscript besides MetaPro-ID specifies the variation of MetaPro-ID with a different encoder.
[12] The model can achieve convergence within 20 epochs.

Table 1
VU Amsterdam Metaphor Corpus dataset statistics. OC is open-class word annotated. AP is all-PoS annotated. # tgt token is the number of target tokens whose metaphoricity is to be identified. % m is the percentage of metaphoric tokens among target tokens. # seq is the number of sequences. Len seq is the average length of sequences.

		# tgt token	% m	# seq	Len seq
VUA-OC	Train	57,799	15.2	8,716	16.3
	Valid	14,812	15.4	2,178	16.6
	Test	22,196	17.9	3,698	15.5
	All	94,807	15.9	14,592	16.1
VUA-AP	Train	116,622	11.2	6,323	18.4
	Valid	38,628	11.6	1,550	24.9
	Test	50,175	12.4	2,694	18.6
	All	205,425	11.6	10,567	19.4

Table 2
Formal Idioms Corpus dataset statistics. # uniq MWEs is the number of unique MWEs. Len MWE is the average length of the unique MWEs.

	# uniq MWEs	Len MWEs	# seq	Len seq
FIC	358	2.6	3136	25.9

the probabilities of positive, negative, and neutral polarities for an input sequence.

We use the default setups of the above APIs. Vader and Azure are domain non-specific. AllenNLP is a cross-domain classifier for our examined news headline sentiment analysis task. We include another RoBERTa-based sequence classification baseline model that is trained and tested with the benchmarking dataset. This RoBERTa classifier is news headline task-specific.

4.3. Datasets

VUA VU Amsterdam Metaphor Corpus (VUA) [3] is the largest all-word annotated metaphor detection dataset. It covers metaphors from different genres, e.g., academic texts, conversation, news, and fiction. The dataset was used with different annotation paradigms. Metaphor Detection Shared Task 2018 [41] and 2020 [42] used the VUA dataset for learning open-class (OC) metaphor detection, where only the target words are evaluated. Gao et al. [12] prepared another widely used VUA benchmark dataset, considering all tokens in a sequence as target tokens. This dataset covers metaphors in all-PoS (AP), used by Mao et al. [13], Dankers et al. [53], Le et al. [15] and Mao and Li [16]. We evaluate our MetaPro-ID model on these two datasets (VUA-OC[13] and VUA-AP[14]). Since the VUA-OC dataset does not have a validation set, we randomly split 20% of the training sentences as the validation set. We also randomly select 100 sentences from the VUA-OC testing set, containing 266 metaphors to evaluate the metaphor paraphrasing module. The detailed statistics can be viewed in Table 1.[15]

FIC Formal Idioms Corpus[16] (FIC) was collected by Saxena and Paul [65] in their English Possible Idiomatic Expressions (EPIE) corpus. It covers idioms with various lexical modifications. We use FIC to test our dictionary and rule-based method in detecting MWEs with different variations. Each idiomatic lemma has 8.8 associated sentences on average. The dataset is pre-processed with lowercase. The detailed statistics can be viewed in Table 2.

NHSA News headline sentiment analysis (NHSA) dataset was collected from the financial domain in SemEval 2017 Task 5 [32]. We use NHSA to test MetaPro text pre-processing, based on different sentiment analysis classifiers. The original dataset[17] was labeled with numerical sentiment scores, ranging from −1 to 1. For a fair comparison between different classifiers, we binarize the scores into positive and negative sentiment polarities. We exclude 50 instances with 0 scores, because the sentiment analysis API from AllenNLP can only yield positive and negative predictions. Besides, if the predicted positive probability is higher than the predicted negative probability, the prediction is positive in Vader and Azure, otherwise, negative. Table 3 shows the statistics of NHSA.

Table 3
News headline sentiment analysis dataset statistics. % pos is the percentage of positive sequences among all sequences. % neg is the percentage of negative sequences among all sequences.

	# seq	% pos	% neg	Len seq
NHSA	1597	58.5	41.5	9.6

Table 4
Metaphor identification performance on the VUA-open-class testing set.

Model	P	R	F1	Acc
Wu et al. [11]	61.1	67.7	64.3	–
Su et al. [14]	72.8	72.6	72.7	90.2
RoBERTa	74.1	68.5	71.2	90.2
RoBERTa+PoS	74.2	69.1	71.6	90.3
MetaPro-ID	74.3	71.9	**73.1***	**90.5***

*Denotes the improvement is statistically significant, based on a 2-tailed test ($p < 0.05$).

Table 5
Metaphor identification performance on the VUA-all-PoS testing set.

Model	P	R	F1	Acc
Gao et al. [12]	71.6	73.6	72.6	93.1
Mao et al. [13]	73.0	75.7	74.3	93.8
Dankers et al. [53]	–	–	76.9	–
Le et al. [15]	74.8	75.5	75.1	93.8
RoBERTa	78.7	75.6	77.1	94.3
RoBERTa+PoS	78.7	75.9	77.3	94.4
MetaPro-ID$_{BERT}$	78.3	76.9	**77.6***	**94.5***
MetaPro-ID	80.9	77.6	**79.2***	**94.9***

5. Results

We evaluate the sub-modules of MetaPro on metaphor identification (Section 5.1) and interpretation (Section 5.2) tasks, respectively. The evaluation of overall performance of MetaPro is based on a downstream sentiment analysis task in Section 5.3.

5.1. Metaphor identification

We benchmark the metaphor identification module of MetaPro (Metaphor-ID) on the VUA-OC dataset and VUA-AP dataset, respectively. The performance is measured by F1 score, where metaphors are positive labels.

As seen in Tables 4 and 5, MetaPro-ID surpasses all the baselines. Compared with the strongest external baseline in each dataset, MetaPro-ID achieves an average gain of 1.3% F1 scores. Specifically, Su et al. [14] achieves 72.7% F1 on VUA-OC; Dankers et al. [53] achieves 76.9% F1 on VUA-AP; MetaPro achieves 73.1% F1 on VUA-OC and 79.2% F1 on VUA-AP, respectively. Our BERT-based model (MetaPro-ID$_{BERT}$, 77.6%) also outperforms the strongest BERT-based baseline

[13] Based on the toolkit from https://github.com/EducationalTestingService/metaphor/tree/master/VUA-shared-task Accessed 2 September 2020.

[14] Originally provided by https://github.com/gao-g/metaphor-in-context Accessed 6 November 2018.

[15] We exclude the sentences that do not have a target word in VUA-OC dataset.

[16] Originally provided by https://github.com/prateeksaxena2809/EPIE_Corpus Accessed 8 February 2021.

[17] Originally provided by https://alt.qcri.org/semeval2017/task5/index.php?id=data-and-tools Accessed 15 February 2021.

Table 6
Metaphor identification performance on different types of open-class words in the VUA-open-class testing set.

	Model	P	R	F1	Acc
VERB	Su et al. [14]	73.8	77.3	75.5	85.0
	MetaPro-ID^{AP}	78.8	74.5	76.6	**86.2**
	MetaPro-ID^{OC}	77.3	76.7	**77.0**	**86.2**
NOUN	Su et al. [14]	74.3	69.8	**72.0**	92.7
	MetaPro-ID^{AP}	76.8	61.9	68.5	92.4
	MetaPro-ID^{OC}	76.6	66.9	71.4	**92.8**
ADJ	Su et al. [14]	70.2	69.9	70.1	89.0
	MetaPro-ID^{AP}	73.1	60.5	66.2	88.6
	MetaPro-ID^{OC}	71.6	69.8	**70.7**	**89.3**
ADV	Su et al. [14]	58.7	58.3	58.5	92.8
	MetaPro-ID^{AP}	68.9	61.8	65.2	**94.3**
	MetaPro-ID^{OC}	65.9	66.9	**66.4**	94.2

[53] (76.9%) by 0.7% F1 on the VUA-AP dataset. Using RoBERTa pre-trained Language Model instead of BERT further boosts the model performance by 1.6% on VUA-AP. Compared with using a vanilla RoBERTa sequence tagging model, our MTL and Gated Bridging Mechanism-based model yields an average gain of 2.0% on the two datasets.[18] Compared with RoBERTa+PoS, MetaPro-ID yields an average gain of 1.7% F1 over the two datasets. This shows that the MTL-based MetaPro-ID has better feature fusion capacity than the single task learning-based RoBERTa+PoS model. Besides, MTL takes the advantage of reducing the risk of overfitting [52], compared with single-task learning. Thus, MetaPro-ID yielding extra gains. The above improvements show that MetaPro-ID is the state-of-the-art model in sequential metaphor identification.

In the breakdown analysis of open-class metaphor identification, we benchmark with the work of Su et al. [14], because it is the strongest baseline on the VUA-OC dataset. We are also interested in the performance of models that are trained on different datasets with different annotation paradigms (all-PoS and open-class word annotations). Thus, MetaPro-ID that was trained on VUA-AP dataset and VUA-OC dataset is also introduced for benchmarking, termed as MetaPro-ID^{AP} and MetaPro-ID^{OC}. As seen in Table 6, MetaPro-ID^{OC} achieves the highest F1 scores on verbs, adjectives, and adverbs among all the baselines. In the comparison between models that were trained on different datasets, MetaPro-ID^{OC} yields higher F1 scores on the four PoS than MetaPro-ID^{AP}. In light of this, we embed the model that was trained on the VUA-OC dataset into MetaPro.

5.2. Metaphor interpretation

We compare the performance of our proposed RoBERTa-based metaphor paraphrasing method to BERT and ALBERT-based methods. These methods follow a similar framework that uses Language Models and WordNet. We do not benchmark with other external metaphor interpretation baselines, because as argued in Section 2 (metaphor interpretation related works), these methods cannot be directly used in everyday texts from end-to-end. The applications of these methods depend on hand-coded knowledge in a specific application scenario, a specific PoS, or a specific concept domain. However, our metaphor interpretation evaluation data (VUA) is natural language from four different genres (academic texts, conversation news, and fiction) with different open-class PoS and concepts.

[18] The detailed ablation analysis of using multi-task learning, different information transformation mechanisms, weighted sum pooling strategy, and the effectiveness of Gated Bridging Mechanism in the fusing and filtering of information between the multi-task learning towers of metaphor identification and PoS tagging can be viewed in the work of Mao and Li [16].

Due to the absence of gold labels of metaphor paraphrases, we cannot use automatic evaluation matrices, such as BLEU [87] and ROUGE [88]. Thus, the paraphrases are evaluated by humans. For evaluating the coverage and accuracy of our dictionary and rule-based metaphoric MWE detection method, we introduce Machine Learning baselines (BiLSTM-CRF and RoBERTa) and an external idiomatic dataset (FIC) that has diverse variations in idiom word forms and contexts.

5.2.1. Preliminary window size evaluation and human evaluation introduction

We first identify the optimal window size for the paraphrasing module of MetaPro. We sample 40 unique target metaphors (non-MWEs) from their associated 40 long sentences (the length of a sentence is greater than 30 tokens) in the VUA-OC validation set. Each PoS (verbs, nouns, adjectives, and adverbs) has 10 different metaphoric target words. The average length of the selected sentences is 45.8 tokens. We set up five types of windows sizes (window = {4, 8, 12, 16, none}). None-window means that all context words in a sequence are used for predicting the paraphrase of the target metaphor.

We evaluate the quality of a paraphrased word from three dimensions with three questions in human evaluation: Q1. Does the paraphrased word semantically and grammatically fit the context (coherence)? Q2. To what extent does the paraphrased word represent the contextual meaning of the original target word (semantic completeness)? Q3. Is the paraphrased word a literal counterpart of the original target word (literality)? The basic meaning of an appropriate literal counterpart of a target word should be similar to the contextual meaning of the target word. The options of each question are in a 1–5 interval scale, representing very unlikely, unlikely, maybe, likely, and very likely, respectively. Each paraphrase is evaluated by three native English speakers from the UK and the US. The survey was conducted on Amazon Mechanical Turk.[19] We employ Fleiss [89] kappa (κ) as an agreement measure for human evaluations, where $\kappa^{coh.}$ (coherence), $\kappa^{s.c.}$ (semantic completeness) and $\kappa^{lit.}$ (literality) are 0.53, 0.55 and 0.57, respectively. The result is averaged over the number of target metaphors and participants in each evaluation dimension and window size. The overall performance is measured by the average score of the three dimensions.

As seen in Fig. 5a, the optimal window sizes for RoBERTa, ALBERT, and BERT are 16, none, and none, respectively, where RoBERTa outperforms the other two pre-trained Language Models across all the window sizes in this long sentence metaphor interpretation evaluation task. A common trend is that large window size is more appropriate for inferring the paraphrases in long sentences. This is because a larger context can provide more dependent information for the best-fit word prediction in Eqs. (13) and (14). However, we also observe a drop in RoBERTa in using the full context (non-window). It shows that for RoBERTa, a very large context may introduce errors in predicting paraphrases. By viewing Fig. 5b, c, and d, the overall optimal window size also works for each individual evaluation dimension. Thus, we will use these identified window sizes in the following metaphor paraphrase evaluation.

5.2.2. Human metaphor paraphrase evaluation

We conduct another formal human evaluation task for selecting an optimal Language Model to embed it in MetaPro. We randomly sample 100 sentences from the VUA-OC testing set. Each selected sentence has at least a paraphrase for an identified metaphor. MetaPro identifies 249 metaphors (117 verbs, 97 nouns, 24 adjectives, and 11 adverbs), where RoBERTa, ALBERT, and BERT-based paraphrasing modules yield 220, 223, and 218 paraphrases (the human evaluation targets), respectively. Compared with verbs and nouns, the numbers of

[19] https://www.mturk.com Accessed 16 April 2021.

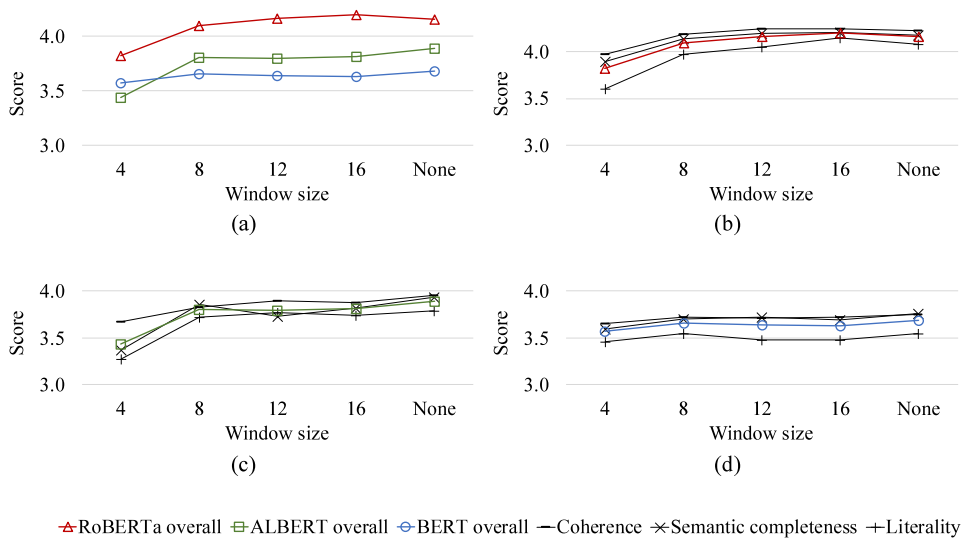

Fig. 5. Window size analysis for metaphor paraphrases in long sentences. The higher score the better. (a) Overall comparison between different pre-trained Language Models. The performance of (b) RoBERTa, (c) ALBERT, and (d) BERT in coherence, semantic completeness and literality evaluation dimensions.

Table 7
Metaphor paraphrase performance.

Model	Coh.	S.C.	Lit.	Avg
ALBERT	4.30	3.91	3.79	4.00
BERT	4.34	4.03	3.77	4.05
RoBERTa	**4.48**	**4.18**	**4.00**	**4.22**

identified adjective and adverbial metaphors are much lower in the 100 random sentences. This is because adjective and adverbial metaphors are less common in our applied dataset, e.g., adjective and adverbial metaphors only take 14.9% and 3.4% of all metaphors in the testing set, respectively. Besides, the numbers of paraphrases that are given by different Language Models are lower than the number of identified metaphors. This is because, if the probability of the best-fit word of a metaphor is not one of the top u ($u = 5000$) possible replacements, the identified metaphor is not paraphrased. We will test the sensitivity of u with different values on a sentiment analysis task later. We use the same evaluation criteria that were mentioned in Section 5.2.1 (three questions for three evaluation dimensions, and three native English speaker annotators for each question). $\kappa^{coh.}$, $\kappa^{s.c.}$ and $\kappa^{lit.}$ are 0.54, 0.51 and 0.56, respectively in this human evaluation task.

As seen in Table 7, RoBERTa achieves better performance across all the evaluation dimensions, yielding a gain of 0.17 on the average score against BERT. Although ALBERT is lighter than the other two models, the performance is weaker. Thus, we embed RoBERTa-based metaphor paraphrasing model into MetaPro.

In the individual evaluation dimension, the common trend is that each Language Model achieves higher scores in coherence, while the scores in literality are comparatively lower. This is because the selected Language Models can easily yield coherent missing words by training with large corpora, while a paraphrase is still possibly metaphoric. This is likely because conventional metaphors have been commonly appearing in everyday language. Thus, these metaphors also have high probabilities of appearing in a context during the metaphor paraphrasing procedure.

In breakdown analysis in Table 8, we observe that RoBERTa takes advantage across all the open-class PoS. Compared RoBERTa to other Language Models, the largest gain in average scores appears in adverbs (0.55). Overall, the RoBERTa-based method yields acceptable paraphrases, measured by the average score (above 3.98) across all the open-class PoS.

Table 8
Metaphor interpretation performance on different types of open-class words.

	Model	Coh.	S.C.	Lit.	Avg
VERB	ALBERT	4.24	4.02	3.85	4.04
	BERT	4.35	4.14	3.87	4.12
	RoBERTa	**4.55**	**4.28**	**4.07**	**4.30**
NOUN	ALBERT	4.38	3.89	3.79	4.02
	BERT	4.34	3.95	3.71	4.00
	RoBERTa	**4.41**	**4.11**	**3.97**	**4.16**
ADJ	ALBERT	4.31	3.56	3.46	3.78
	BERT	4.30	3.72	3.72	3.91
	RoBERTa	**4.33**	**3.8**	**3.8**	**3.98**
ADV	ALBERT	4.06	3.72	3.44	3.74
	BERT	4.27	3.80	3.13	3.73
	RoBERTa	**4.67**	**4.27**	**3.93**	**4.29**

Table 9
Idiomatic multi-word expression pairing performance, measured by accuracy.

Setup	Acc
Dependency triple pairing only (R1-R3)	88.9
Lemma pairing only (R2-R3)	91.6
Dependency and lemma pairing (R1-R2-R3)	**98.5**

5.2.3. *Metaphoric multi-word expression pairing evaluation*

We test our dictionary and rule-based MWE detection method, based on an idiom dataset (FIC). The performance is measured by accuracy, where the accuracy is given by the number of correctly identified MWEs above the total number of MWEs on token-level.

As seen in Table 9, simply using dependency triple pairing features (R1-R3, 88.9%) and lemma pairing features (R2-R3, 91.6%) achieves 90.3% accuracy on average. The accuracy is improved to 98.5% by combining the two pairing methods. As argued before, there are limitations in each pairing method. However, their combination takes the complementary advantage, thus, yielding better performance. The high accuracy of the combination method also shows that our method has good coverage in identifying idiomatic MWEs with diverse modifications.

Next, we compare the combination of dependency and lemma pairing methods (R1-R2-R3) to sequence tagging models, namely BiLSTM-CRF that was reported by Saxena and Paul [65] along with the FIC dataset, and RoBERTa. Following Saxena and Paul [65], the training and testing sets are the splits of 75% and 25% of the full FIC. We

Table 10
Token-level idiomatic MWE detection performance.

Method	Seen		Unseen	
	Macro-F1	Acc	Macro-F1	Acc
BiLSTM-CRF	93.0	98.0	70.2	92.1
RoBERTa	**97.5**	**99.1**	85.5	95.3
Ours (R1-R2-R3)	94.4	98.7	**97.9***	**99.4***

develop two types of testing sets. The seen case testing set includes the idioms that also appear in the contexts of the training set, while the testing set with unseen cases contains idioms that never appear in the training set. Saxena and Paul [65] formalized the idiom detection task as a sequence tagging task, where the labels employ the B-I-O annotation paradigm, denoting the beginning, the inside, and the outside of an idiomatic expression, respectively. The accuracy is measured by the number of correctly identified idiomatic tokens above the total number of tokens in the testing sets. We report the highest Macro-F1 score and accuracy for the baseline models on the testing set after 20 epoch training.

As seen in Table 10, our method achieves comparable performance against the Machine Learning-based methods on the seen case evaluation at the token level. 86.9% errors of our method are due to the annotation difference between the FIC dataset and our MWE dictionary, which should not be problematic in practice. For example, Saxena and Paul [65] annotated the chunk of "is on cloud nine" as an idiom in "Niall is on cloud nine", while the chunk of "on cloud nine" is defined as an idiom in our dictionary. For the unseen case evaluation, we observe a sharp decrease in both Machine Learning models (more than −12.0% Macro-F1), while the performance of our method is slightly improved (+5.5% Macro-F1). This is because Machine Learning-based methods are also challenged by insufficient coverage of the training set, although RoBERTa (−12.0% Macro-F1) is more generalizable than the BiLSTM-CRF model of Saxena and Paul [65] (−22.8% Macro-F1). On the other hand, idiomatic expressions are more conventional, having similar meanings and pragmatics in contexts, compared with other figurative languages [90]. Our MWE dictionary has covered the vocabularies of the largest idiom dictionary (The Idioms) and dataset [73]. Thus, the performance of our dictionary and rule-based method on the FIC seen and unseen case testing sets are similar. Considering (a) the computational cost of embedding another RoBERTa model that has 355,364,869 parameters in processing MWEs, and (b) the gap between identifying MWE tokens and pairing the identified tokens with the meanings of MWEs in the RoBERTa model, we employ the dictionary and rule-based MWE processing method in MetaPro.

5.3. Metaphor processing for sentiment analysis

We examine MetaPro on a sentiment analysis task, because metaphor understanding is considered as a challenge of achieving human-like sentiment analysis [91]. To achieve better performance on this downstream task, we retrain MetaPro-ID on the combination of VUA-OC training and validation sets. The new model yields 74.0% F1 on the VUA-OC testing set. A news headline dataset (NHSA) from SemEval-2017 Task 5 [32] is used for sentiment analysis evaluation, because news headlines are likely to use metaphors to express rich emotional information, while keeping the language concise [92]. Thus, MetaPro can be more helpful in this scenario. We do not use a metaphor sentiment analysis dataset, because we mean to conduct an unbiased evaluation by keeping the distribution of metaphors in our testing set similar to the metaphor distribution in real-world texts. Furthermore, current sentiment analysis models have achieved accurate performance on many other types of texts [93,94], e.g., the RoBERTa-based sentiment analysis API provided by AllenNLP achieved 95.11% accuracy on the testing set of Stanford Sentiment Treebank [86], where the texts

Table 11
Sentiment analysis results, given by domain non-specific and cross-domain APIs.

	API	Pos. label = Pos.			Pos. label = Neg.			Avg	Acc
		P	R	F1	P	R	F1	F1	
Vader	Original	54.2	43.1	48.0	37.7	48.6	42.5	45.3	45.4
	MetaPro	55.7	45.0	49.8	39.0	49.6	43.7	46.8	46.9
	Gain	+1.5	+1.9	+1.8	+1.3	+1.0	+1.2	+1.5	+1.5
Allen.	Original	86.2	52.8	65.5	57.0	88.1	69.2	67.4	67.4
	MetaPro	87.2	56.2	68.4	58.9	88.4	70.7	69.6	69.6
	Gain	+1.0	+3.4	+2.9	+1.9	+0.3	+1.5	+2.2	+2.2
Azure	Original	69.3	48.7	57.2	49.0	69.5	57.5	57.4	57.4
	MetaPro	69.7	79.7	74.4	64.2	51.3	57.0	65.7	67.9
	Gain	+0.4	+31.0	+17.2	+15.2	−18.2	−0.5	+8.4	+10.5
Avg gain		+1.0	+12.1	+7.3	+6.1	−5.6	+0.7	+4.0	+4.7

are from movie reviews. However, the API achieves 67.4% accuracy on our applied news headline dataset. It shows that news headlines are more challenging for the AllenNLP sentiment analysis API. Thus, potentially, we can improve news headline sentiment analysis with metaphor processing in the scenario that a given classifier is domain non-specific (Vader and Azure), or cross-domain (AllenNLP).

As seen in Table 11, MetaPro achieves 4.7% extra gains in accuracy and 4.0% gains in average F1 score across the three APIs on average. The largest improvement appears in Azure (+8.4% average F1 and +10.5% accuracy). This is because MetaPro fixes many Azure false negative predictions whose true labels are positive (+31.0% recall, given positive polarity as positive labels), although MetaPro also introduces incorrect positive predictions for those negative instances (−18.2% recall, given negative polarity as positive labels). Besides, MetaPro yields extra gains in both rule-based (+1.5% F1 and +1.5% accuracy in Vader) and deep learning-based (+2.2% F1 and +2.2% accuracy in AllenNLP) methods. The above observations demonstrate that metaphoric languages are challenging for news headline sentiment analysis with domain non-specific and cross-domain classifiers. Turning the metaphors into literal languages with MetaPro can somewhat address the issue.

Given a news headline, "Rio Tinto CEO Sam Walsh rejects fears over China growth, demand", the three APIs incorrectly classify the news headline as negative. This is probably because "fears" conveys negative sentiment and "rejects" likely associate negative contexts. However, MetaPro detects these words are metaphoric. The news headline is paraphrased as "Rio Tinto CEO Sam Walsh eliminates concerns over China growth, demand". With the paraphrased input, all the APIs can correctly yield positive predictions. Similar improvements can be observed in interpreting metaphoric MWEs as well, e.g., "Royal Dutch Shell pulls plug on Arctic exploration" is misidentified as positive before pre-processing, while MetaPro interprets the headline as "Royal Dutch Shell pulls plug on Arctic expedition, where 'pull the plug' means that to kill or discontinue", which can be successfully classified as negative by the three APIs.

On the other hand, MetaPro may introduce errors for the APIs. MetaPro paraphrases the headline "UPDATE 1-Norway's Statoil shakes up top management, replaces CFO" as "UPDATE 1-Norway's Statoil changes up top management, replaces CFO", resulting in the failure of the sentiment classifiers. This is because "changes" cannot fully represent the meaning of "shakes" that upsets the stability of the top management in the context. This type of error shows the limitation of our paraphrasing method. The WordNet hypernym and synonym-based paraphrasing method cannot capture the nuance of a metaphoric meaning, e.g., the emotional information and conceptual inferences in a metaphoric expression. Thus, paraphrases may introduce errors for a sentiment analysis classifier. Another type of error is due to the failure of language modeling. Current missing word prediction-based language modeling methods are sub-optimal for inferring the literal counterpart of a metaphor, because they did not use the information of metaphoric

Table 12
The number of introduced and fixed errors in different sentiment analysis APIs.

	Vader	Allen.	Azure
Introduced errors	37	51	173
Fixed errors	61	85	341

Table 13
Sensitivity tests on different sizes of pools of possible replacements of a metaphor in sentiment analysis. Performance is measured by averaged F1 score.

Top u words	Vader	Allen.	Azure	Avg	Gain	Time
Original	45.3	67.4	57.4	56.7	–	–
$u = 100$	46.8	68.1	64.4	59.8	+3.1	1.0x
$u = 1000$	**46.9**	69.1	64.8	60.3	+3.6	1.1x
$u = 3000$	46.6	69.4	64.7	60.2	+3.5	1.4x
$u = 5000$	46.8	**69.6**	**65.7**	60.7	**+4.0**	1.7x
$u = 7000$	46.7	**69.6**	64.8	60.4	+3.7	1.8x
$u = 9000$	46.8	69.4	64.7	60.3	+3.6	1.8x

Table 14
Sentiment analysis results, given by a news headline sentiment analysis task-specific RoBERTa classifier.

Setup	Pos. label = Pos.			Pos. label = Neg.			Avg	Acc
	P	R	F1	P	R	F1	F1	
Original	91.7	86.8	89.2	86.5	91.4	88.9	89.1	89.0
MetaPro	93.4	88.9	91.1	88.6	93.1	90.8	91.0	91.0
Gain	+1.7	+2.1	+1.9	+2.1	+1.7	+1.9	+1.9	+2.0

words during the Language Model pre-training. We will address these issues in future work. The statistics of introduced and fixed errors for each API can be viewed in Table 12.

A threshold that represents the confidence of a metaphor paraphrase (if top u) was manually defined in Eq. (14) in the metaphor paraphrasing module. Here, we conduct sensitivity tests for different u in the sentiment analysis task. Give $u \in \{100, 1000, 3000, 5000, 7000, 9000\}$, we observe that with the growth of u, the accuracy of sentiment analysis can be further improved before $u < 7000$ in Table 13. This is because a metaphor is more likely to be paraphrased with a larger u. However, there is a slight drop in $u = 7000$ and $u = 9000$, respectively. This shows that the accuracy of metaphor processing decreases in sentiment analysis, if paraphrases are less reliable (low probability in context co-occurrence). Besides, a big u also takes more time for searching the best-fit word (see the column of time in Table 13). In practice, one may choose an appropriate u to balance the trade-off between accuracy and time costs.

Another possible application scenario is to employ MetaPro as a text pre-processing method before model training. We examine the utility of MetaPro pre-processing, based on a RoBERTa sequence classification model and NHSA dataset. We split 10% of the official NHSA training set of SemEval 2017 Task 5 as the development set. The testing set (365 sequences, where 52.1% are positive and 47.9% are negative) is in line with the task. The model is trained and tested on the two versions of the NHSA dataset, namely original and MetaPro pre-processed datasets, respectively. In this case, the trained RoBERTa model (either trained with metaphoric texts or paraphrased literal texts) is news headline sentiment analysis task-specific. The reported performance in Table 14 is given by the model that achieves the highest accuracy on the validation set by training 20 epochs.

As seen in Table 14, the classifier trained with MetaPro pre-processing texts delivers 1.9% extra gains on average F1 and 2.0% gains in accuracy, compared with the model trained with the original texts. It shows that MetaPro is also supportive for the task-specific classifier.

6. Conclusion

In this paper, we propose the first end-to-end English metaphor processing model, termed MetaPro, which can be used as a text pre-processing method. MetaPro embeds two modules, namely metaphor identification and metaphor interpretation. Give an input text sequence, the metaphor identification module identifies metaphors on token-level. This is achieved by using a multi-task learning model with a novel information transformation mechanism. We employ Gated Bridging Mechanism [16] for soft-parameter sharing between the sub-task towers, jointly learning metaphor identification and PoS tagging. Next, the metaphor interpretation module can interpret the identified metaphors. If an identified metaphor is an MWE, it is explained via a dictionary and rule-based method. Otherwise, the metaphor is paraphrased as its literal counterpart by using a Language Model and WordNet hypernyms and synonyms of the metaphor. The output of MetaPro is natural language, where the single-word metaphors are replaced with their paraphrases, and the metaphoric MWEs are explained by their dictionary meanings via a clause, beginning with "where".

We examine MetaPro on metaphor identification and interpretation tasks, yielding state-of-the-art performance in both cases. Our dictionary and rule-based idiomatic MWE detection method also delivers sufficient coverage and accuracy, compared with Machine Learning baselines, based on an idiom dataset with various modifications. Finally, we extensively examine the output of MetaPro on a sentiment analysis downstream task. The experimental results show that MetaPro can improve three publicly available sentiment analysis APIs and a strong task-specific classifier on a news headline sentiment analysis task. The improvement can be observed in rule-based, deep learning-based and commercial sentiment classification approaches.

Finally, we find that our method can be further optimized by modeling emotional information and improving the accuracy of a Language Model to better support sentiment analysis. Though our method cannot interpret metaphors in proper nouns and interjections, these metaphors can still be identified. We will study these areas in future work.

CRediT authorship contribution statement

Rui Mao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft. **Xiao Li:** Methodology, Formal analysis, Data curation, Writing – review & editing, Visualization. **Mengshi Ge:** Formal analysis, Data curation, Writing – review & editing. **Erik Cambria:** Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research/project is supported by A*STAR under its Industry Alignment Fund, Singapore (LOA Award I1901E0046).

References

[1] G. Lakoff, The Contemporary Theory of Metaphor, UC Berkeley, 1993.
[2] G. Lakoff, Master Metaphor List, University of California, 1994.
[3] G.J. Steen, A.G. Dorst, J.B. Herrmann, A. Kaal, T. Krennmayr, T. Pasma, A Method for Linguistic Metaphor Identification: From MIP to MIPVU, Vol. 14, John Benjamins Publishing, 2010.
[4] S.M. Mohammad, E. Shutova, P.D. Turney, Metaphor as a Medium for Emotion: An Empirical Study, The *SEM 2016 Organizing Committee, 2016, p. 23.
[5] R. Mao, C. Lin, F. Guerin, Word embedding and WordNet based metaphor identification and interpretation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol. 1, 2018, pp. 1222–1231.

[6] R. Mao, C. Lin, F. Guerin, Interpreting verbal metaphors by paraphrasing, 2021, arXiv preprint arXiv:2104.03391.

[7] E. Cambria, Y. Song, H. Wang, N. Howard, Semantic multi-dimensional scaling for open-domain sentiment analysis, IEEE Intell. Syst. 29 (2) (2014) 44–51.

[8] W. Zhao, H. Peng, S. Eger, E. Cambria, M. Yang, Towards scalable and reliable capsule networks for challenging NLP applications, in: ACL, 2019, pp. 1549–1559.

[9] N. Howard, E. Cambria, Intention awareness: Improving upon situation awareness in human-centric environments, Human-Centric Comput. Inf. Sci. 3 (9) (2013).

[10] E. Shutova, Design and evaluation of metaphor processing systems, Comput. Linguist. 41 (4) (2015) 579–623.

[11] C. Wu, F. Wu, Y. Chen, S. Wu, Z. Yuan, Y. Huang, Neural metaphor detecting with CNN-LSTM model, in: Proceedings of the Workshop on Figurative Language Processing, 2018, pp. 110–114.

[12] G. Gao, E. Choi, Y. Choi, L. Zettlemoyer, Neural metaphor detection in context, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 607–613.

[13] R. Mao, C. Lin, F. Guerin, End-to-end sequential metaphor identification inspired by linguistic theories, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Long Papers), 2019, pp. 3888–3898.

[14] C. Su, F. Fukumoto, X. Huang, J. Li, R. Wang, Z. Chen, DeepMet: A reading comprehension paradigm for token-level metaphor detection, in: Proceedings of the 2nd Workshop on Figurative Language Processing, 2020, pp. 30–39.

[15] D. Le, M. Thai, T. Nguyen, Multi-task learning for metaphor detection with graph convolutional neural networks and word sense disambiguation, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 8139–8146.

[16] R. Mao, X. Li, Bridging towers of multitask learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 13534–13542.

[17] E. Shutova, D. Kiela, J. Maillard, Black holes and white rabbits: Metaphor identification with visual features, in: Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 160–170.

[18] M. Rei, L. Bulat, D. Kiela, E. Shutova, Grasping the finer point: A supervised similarity network for metaphor detection, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1537–1546.

[19] E. Shutova, T. Van de Cruys, A. Korhonen, Unsupervised metaphor paraphrasing using a vector space model, in: 24th International Conference on Computational Linguistics, 2012, p. 1121.

[20] D. Bollegala, E. Shutova, Metaphor interpretation using paraphrases extracted from the web, PLoS One 8 (9) (2013).

[21] H. Li, K.Q. Zhu, H. Wang, Data-driven metaphor recognition and explanation, Trans. Assoc. Comput. Linguist. 1 (2013) 379–390.

[22] C. Su, S. Huang, Y. Chen, Automatic detection and interpretation of nominal metaphor based on the theory of meaning, Neurocomputing 219 (2017) 300–311.

[23] S. Narayanan, Knowledge-Based Action Representations for Metaphor and Aspect (KARMA), Computer Science Division, University of California at Berkeley Dissertation, 1997.

[24] J. Martin, A Computational Model of Metaphor Interpretation, Academic Press Professional, 1990.

[25] J.A. Barnden, M.G. Lee, An artificial intelligence approach to metaphor understanding, Theor. Hist. Sci. 6 (1) (2002) 399–412.

[26] C. Widera, T. Portele, M. Wolters, Prediction of word prominence, in: Proceedings of the 5th European Conference on Speech Communication and Technology, 1997, pp. 999–1002.

[27] N.R. Norrick, Interjections, in: Pragmatics of Society, Mouton de Gruyter Berlin, 2011, pp. 243–292.

[28] A. Vicente, Polysemy and word meaning: An account of lexical meaning for different kinds of content words, Philos. Stud. 175 (4) (2018) 947–968.

[29] S. Feng, E. Wallace, A. Grissom II, P. Rodriguez, M. Iyyer, J. Boyd-Graber, Pathologies of neural models make interpretation difficult, in: Empirical Methods in Natural Language Processing, 2018, pp. 3719–3728.

[30] R.W. Gibbs, Why idioms are not dead metaphors, in: Idioms: Processing, Structure, and Interpretation, Erlbaum Hillsdale, 1993, pp. 57–77.

[31] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009.

[32] K. Cortis, A. Freitas, T. Daudert, M. Huerlimann, M. Zarrouk, S. Handschuh, B. Davis, SemEval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news, in: Proceedings of the 11th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2017, pp. 519–535.

[33] Y. Wilks, A preferential, pattern-seeking, semantics for natural language inference, Artificial Intelligence 6 (1) (1975) 53–74.

[34] Y. Wilks, Making preferences more active, Artificial Intelligence 11 (3) (1978) 197–223.

[35] M. Black, et al., More about metaphor, in: Metaphor and Thought, Vol. 2, 1979, pp. 19–41.

[36] G. Lakoff, M. Johnson, Metaphors We Live by, University of Chicago Press, 1980.

[37] E. Semino, G. Steen, Metaphor in literature, in: The Cambridge Handbook of Metaphor and Thought, Cambridge University Press, Cambridge, 2008, pp. 232–246.

[38] G. Pragglejaz, MIP: A method for identifying metaphorically used words in discourse, Metaphor Symb. 22 (1) (2007) 1–39.

[39] Z. Kovecses, Metaphor: A Practical Introduction, Oxford University Press, 2010.

[40] M. Ge, R. Mao, E. Cambria, Explainable metaphor identification inspired by conceptual metaphor theory, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022.

[41] C.W.B. Leong, B.B. Klebanov, E. Shutova, A report on the 2018 VUA metaphor detection shared task, in: Proceedings of the Workshop on Figurative Language Processing, 2018, pp. 56–66.

[42] C.W. Leong, B.B. Klebanov, C. Hamill, E. Stemle, R. Ubale, X. Chen, A report on the 2020 VUA and TOEFL metaphor detection shared task, in: Proceedings of the 2nd Workshop on Figurative Language Processing, 2020, pp. 18–29.

[43] J. Birke, A. Sarkar, A clustering approach for nearly unsupervised recognition of nonliteral language, in: 11th Conference of the European Chapter of the Association for Computational Linguistics, 2006, pp. 329–336.

[44] I. Heintz, R. Gabbard, M. Srinivasan, D. Barner, D.S. Black, M. Freedman, R. Weischedel, Automatic extraction of linguistic metaphor with LDA topic modeling, in: Proceedings of the 1st Workshop on Metaphor in NLP, 2013, pp. 58–66.

[45] D. Hovy, S. Shrivastava, S.K. Jauhar, M. Sachan, K. Goyal, H. Li, W. Sanders, E. Hovy, Identifying metaphorical word use with tree kernels, in: Proceedings of the 1st Workshop on Metaphor in NLP, 2013, pp. 52–57.

[46] W. Song, S. Zhou, R. Fu, T. Liu, L. Liu, Verb metaphor detection via contextual relation learning, 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), 2021, pp. 4240–4251.

[47] M. Choi, S. Lee, E. Choi, H. Park, J. Lee, D. Lee, J. Lee, MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 1763–1773.

[48] R. Mao, C. Lin, F. Guerin, Combining pre-trained word embeddings and linguistic features for sequential metaphor identification, 2021, arXiv preprint arXiv:2104.03285.

[49] D. Fass, Met*: A method for discriminating metonymy and metaphor by computer, Comput. Linguist. 17 (1) (1991) 49–90.

[50] H. Wan, J. Lin, J. Du, D. Shen, M. Zhang, Enhancing metaphor detection by gloss-based interpretations, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1971–1981.

[51] M. Honnibal, I. Montani, SpaCy 2: Natural language understanding with bloom embeddings, in: Convolutional Neural Networks and Incremental Parsing, 2017.

[52] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, arXiv preprint arXiv:1706.05098.

[53] V. Dankers, M. Rei, M. Lewis, E. Shutova, Modelling the interplay of metaphor and emotion through multitask learning, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 2218–2229.

[54] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, arXiv preprint arXiv:1907.11692.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[56] N.F. Liu, M. Gardner, Y. Belinkov, M.E. Peters, N.A. Smith, Linguistic knowledge and transferability of contextual representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 1073–1094.

[57] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.

[58] L. Cameron, Metaphor in Educational Discourse, A&C Black, 2003.

[59] J.H. Martin, A corpus-based analysis of context effects on metaphor comprehension, Trends Linguist. Stud. Monogr. 171 (2006) 214.

[60] C. Fellbaum, WordNet: An Electronic Lexical Database, Bradford Books, 1998.

[61] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[62] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

[63] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, in: Proceedings of 18th International Conference on Learning Representations, 2020, pp. 1–17.

[64] B. Santorini, Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision), Technical Reports (CIS), 1990, p. 570.
[65] P. Saxena, S. Paul, EPIE dataset: A corpus for possible idiomatic expressions, in: International Conference on Text, Speech, and Dialogue, Springer, 2020, pp. 87–94.
[66] O. Rohanian, M. Rei, S. Taslimipoor, et al., Verbal multiword expressions for identification of metaphor, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2890–2895.
[67] R. Sun, H. Cui, K. Li, M.-Y. Kan, T.-S. Chua, Dependency relation matching for answer selection, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 651–652.
[68] H. Cui, R. Sun, K. Li, M.-Y. Kan, T.-S. Chua, Question answering passage retrieval using dependency relations, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 400–407.
[69] D.M. Bikel, V. Castelli, Event matching using the transitive closure of dependency relations, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, 2008, pp. 145–148.
[70] M. Constant, G. Eryiğit, J. Monti, L. Van Der Plas, C. Ramisch, M. Rosner, A. Todirascu, Multiword expression processing: A survey, Comput. Linguist. 43 (4) (2017) 837–892.
[71] M. Carpuat, M. Diab, Task-based evaluation of multiword expressions: A pilot study in statistical machine translation, in: Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 242–245.
[72] M. Ghoneim, M. Diab, Multiword expressions in the context of statistical machine translation, in: Proceedings of the 6th International Joint Conference on Natural Language Processing, 2013, pp. 1181–1187.
[73] R. Agrawal, V.C. Kumar, V. Muralidharan, D.M. Sharma, No more beating about the bush: A step towards idiom handling for Indian language NLP, in: Proceedings of the 11th International Conference on Language Resources and Evaluation, 2018, pp. 319–324.
[74] S. Glucksberg, M.S. McGlone, Understanding Figurative Language: From Metaphor to Idioms, 36, Oxford University Press on Demand, 2001.
[75] M.-L. Pitzl, Creativity, idioms and metaphorical language in ELF, in: Routledge Handbook of English as a Lingua Franca, Vol. 233, Routledge London, 2017, p. 243.
[76] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.
[77] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[78] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[79] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (5–6) (2005) 602–610.
[80] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, 2018, pp. 2227–2237.
[81] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.
[82] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv preprint arXiv:1609.08144.
[83] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, 2015, arXiv preprint arXiv:1508.01991.
[84] J.D. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 282–289.
[85] C. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 8, 2014, pp. 216–225.
[86] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[87] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
[88] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Proceedings of the ACL-04 Workshop: Text Summarization Branches Out, Association for Computational Linguistics, 2004, pp. 74–81.
[89] J.L. Fleiss, Measuring nominal scale agreement among many raters., Psychological Bulletin 76 (5) (1971) 378.
[90] R.W. Gibbs Jr., N.P. Nayak, Why idioms mean what they do, J. Exp. Psychol. [Gen.] 120 (1) (1991).
[91] E. Cambria, S. Poria, A. Gelbukh, M. Thelwall, Sentiment analysis is a big suitcase, IEEE Intell. Syst. 32 (6) (2017) 74–80.
[92] E. Kitis, M. Milapides, Read it and believe it: How metaphor constructs ideology in news discourse. A case study, J. Pragmat. 28 (5) (1997) 557–590.
[93] E. Cambria, Q. Liu, S. Decherchi, F. Xing, K. Kwok, SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In: LREC (2022).
[94] A. Yadav, D.K. Vishwakarma, Sentiment analysis using deep learning architectures: A review, Artif. Intell. Rev. 53 (6) (2020) 4335–4385.