

# Explainable Metaphor Identification Inspired by Conceptual Metaphor Theory

Mengshi Ge\*, Rui Mao\*, Erik Cambria

Nanyang Technological University, Singapore

mengshi001@e.ntu.edu.sg, rui.mao@ntu.edu.sg, cambria@ntu.edu.sg

## Abstract

Metaphor is not only a linguistic phenomenon but also reflects the concept projection between source and target domains in human cognition. Previous sequence tagging-based metaphor identification methods could not model the concept projection, resulting in a limitation that the outputs of these models are unexplainable in the predictions of the metaphoricity labels. In this work, we propose the first explainable metaphor identification model, inspired by Conceptual Metaphor Theory. The model is based on statistic learning, a lexical resource, and a novel reward mechanism. Our model can identify the metaphoricity on the word-pair level, and explain the predicted metaphoricity labels via learned concept mappings. The use of the reward mechanism allows the model to learn the optimal concept mappings without knowing their true labels. Our method is also applicable for the concepts that are out of training domains by using the lexical resource. The automatically generated concept mappings demonstrate the implicit human thoughts in metaphoric expressions. Our experiments show the effectiveness of the proposed model in metaphor identification, and concept mapping tasks, respectively.

## Introduction

Metaphor is a special linguistic phenomenon, using one or several words to modify a target concept that is different from the source concept. Lakoff and Johnson (1980) proposed a Conceptual Metaphor Theory (CMT), arguing that metaphorical expressions are the linguistic surface realization of metaphorical concepts in human cognitive systems, reflecting human thoughts and behaviors. They explained the cognitive mechanisms of metaphors in the form of the mapping of concepts between source and target domains, e.g., given “you are *wasting*<sup>1</sup> my time”, “*wasting*” is metaphoric, because conceptually, TIME IS MONEY<sup>2</sup> in the context (Lakoff and Johnson 1980). The source concept MONEY and the target concept TIME are domain-different. The metaphoric expression frames TIME in the MONEY shape, associating with precious and scarce attributes.

Thus, the studying of concept mapping mechanisms helps infer the implicit meanings and imageabilities of metaphorical expressions. Furthermore, the conceptualization and mapping also explain why a word is metaphoric with a general description, e.g., TIME IS MONEY also explains the metaphoricity of “your help will *save* me days” and “I have *invested* many months in the project” and, hence, also enables intention mining (Howard and Cambria 2013). Based on CMT, Lakoff, Espenson, and Schwartz (1991) compiled a Master Metaphor List to categorize common concepts and mappings with great labor efforts. However, Shutova and Simone (2010) argued that the abstractness level of the listed concept agents (e.g., TIME and MONEY) was hardly controlled, because of the subjectivity of annotators. Additionally, the listed concept agents failed to represent the whole spectrum of metaphoric expressions, resulting in the fact that the list was rarely used in the computational metaphor community. Finally, current deep learning-based metaphor detection methods are unexplainable, because these methods (Gao et al. 2018; Su et al. 2020) can simply yield the metaphoricity labels for individual tokens. Our motivation is to propose an automatic method to mitigate the labor efforts and the subjective issues in conceptualization; We project concept agents to WordNet (Fellbaum 1998) to gain a broader spectrum and a unified conceptualization criterion for concept generations.

Given a pair of dependent words, “*clean* datum”, our model identifies the word pair as metaphoric, and generating source and target concepts in natural language, e.g., INFORMATION IS MATERIAL. We do not have a very large corpus, containing all metaphoric concepts for the supervised learning. To abstract appropriate concept agents that promote the accuracy of the metaphor identification, we propose a novel dynamic reward mechanism. The reward mechanism allows the model to identify an efficient concept without knowing the true label of the concept. An accurate metaphoricity label prediction will reward the associated source and target concept predictions. Thus, the reward mechanism can push the model to yield more accurate metaphor identifications and effective concept generations. Due to the absence of word pair datasets in other Parts-of-Speech (PoS), we focus on verb-noun and adjective-noun metaphor identification, which is in line with Shutova, Sun, and Korhonen (2010); Shutova, Kiela, and Maillard (2016) and Rei et al. (2017).

\*These authors contributed equally.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

<sup>1</sup>Italics are metaphors.

<sup>2</sup>Capitalization represents concepts and mappings.

In this work, we first demonstrate that our model outperforms previous baselines in word pair-level metaphor identification, yielding an average gain of 3.1% F1 scores on two publicly available datasets (Mohammad, Shutova, and Turney 2016; Tsvetkov et al. 2014). In the automatic evaluation of source and target concept generations, we observe 10.1% gains in F1 over the baseline with random concepts, based on a vanilla RoBERTa classifier (Liu et al. 2019) and a dataset (Gutierrez et al. 2016) from a different domain. Finally, the human evaluation results demonstrate that our method achieves 63.7% average accuracy over three datasets in the concept mapping evaluation task, 70.7% accuracy in the source concept generation task, and 87.3% accuracy in the target concept generation task.

The contribution of this work is twofold: (1) We propose a novel method for explainable metaphor identification. The model is informed by CMT, identifying metaphoricity, and generating source and target concepts for metaphoric word pairs. (2) We demonstrate that our method achieves state-of-the-art performance in word pair metaphor identification. The concept mapping evaluation tasks also show that our model is more accurate than the baselines.

## Related Work

Metaphor identification is a widely studied task in metaphor processing, focusing on detecting metaphors on token-level (Stowe et al. 2019; Mao, Lin, and Guerin 2019; Su et al. 2020; Mao and Li 2021; Choi et al. 2021), relation-level (Zayed, McCrae, and Buitelaar 2020), word pair-level (Shutova, Kiela, and Maillard 2016; Rei et al. 2017), and sentence-level (Birke and Sarkar 2006; Heintz et al. 2013). Currently, sequence-tagging models have achieved significant improvements in linguistic metaphor detection (Leong, Klebanov, and Shutova 2018; Leong et al. 2020). However, these models fail to explain the conceptual mapping mechanism of source and target domains of a metaphoric expression. Investigating the concept mapping mechanisms helps us understand the implicit imageability of a metaphor.

Recent conceptual metaphor processing methods targeted to map source and target word clusters (Mason 2004; Shutova et al. 2017). These methods used word co-occurrence features, and traditional clustering algorithms to identify the metaphoricity of word pairs. However, these methods cannot automatically conceptualize the clusters, generating abstract concept agents in natural language to represent the clusters. There is still a one-step gap to the real-world applications by linguistic learners.

In theoretical studies, Lakoff and Johnson (1980) proposed CMT to explain the concept mapping mechanisms of metaphors. Scholars summarized and generated a list of concept mappings, namely Master Metaphor List (Lakoff, Espenso, and Schwartz 1991). In the list, the concepts were abstracted, grouped, and categorized, according to large-scale corpus studies. However, the listed concepts are not enough for representing all the metaphorical concepts in everyday life (Shutova and Simone 2010), resulting in the fact that these concept mapping instances were rarely used in the computational metaphor processing community.

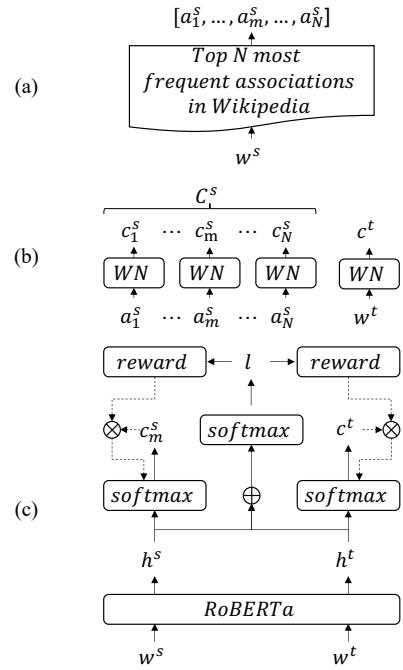


Figure 1: The framework of our model. (a) Common association acquisition. (b) Conceptualization. (c) Metaphor identification and concept generation.  $w$  is an input word.  $s$  and  $t$  denote source and target domains, respectively.  $a_m^s$  denotes one of the top  $N$  most frequent associations of  $w^s$  in the dependency of the word pair  $(w^s, w^t)$  in a Wikipedia dump.  $C^s = \{c_1^s, \dots, c_m^s, \dots, c_N^s\}$  is a set of source candidate concept agents, given by WordNet ( $WN$ ) and a statistical algorithm.  $c^t$  is a generated target concept.  $h$  is a hidden state, given by a RoBERTa encoder.  $l$  is the metaphoricity label of a word pair  $(w^s, w^t)$  input.  $\otimes$  denotes scalar multiplication.  $\oplus$  denotes concatenation. In Figure (c), a solid line denotes forward propagation; a dashed line is backward propagation.

Using considerable human labor to develop such a list also signifies the importance of automatically generating abstract concepts in natural language, and mapping source and target domains for the community. We mean to propose an explainable model to detect metaphors, automatically generating concepts to represent source and target domains with a broad spectrum.

## Methodology

Our model inputs are dependent word pairs, either verb-noun (in a subject-verb or a verb-direct object dependency relationship) or adjective-noun. The noun part is literal, while the verb and adjective parts are either literal or metaphoric. The output label indicates the metaphoricity of a word pair (metaphor or literal). We define the nouns as the words ( $w^t$ ) for inferring target concepts; the verbs and adjectives are the words ( $w^s$ ) for inferring source concepts. Our model (Fig. 1) employs three technical parts to address the issue that current conceptual metaphor corpora cannot cover the whole concept spectrum.

First, we acquire the  $N$  most frequent noun associations ( $[a_1^s, \dots, a_m^s, \dots, a_N^s]$ ) of  $w^s$  in the same dependency of the word pair  $(w^s, w^t)$  from a Wikipedia dump<sup>3</sup>. Next,  $a^s$  and  $w^t$  are conceptualized with WordNet and a statistical knee point algorithm (Satopaa et al. 2011). The abstract concept ( $c_m^s$ ) of  $a_m^s$  is a source candidate concept, where  $c_m^s \in C^s$ . The abstract concept of  $w^t$  is the generated target concept ( $c^t$ ). Projecting a word to a concept in WordNet allows us to handle unseen concepts after training. Finally, the multitask learning (MTL) model learns metaphor detection and conceptualization jointly with a reward mechanism to generate target and optimal source concepts.

## Common Association Acquisition

By observing the examples given by Lakoff and Johnson (1980), we find a common conceptualization pattern in verbal and adjective metaphors. The target concept agents are represented as the abstraction of the dependent nouns of metaphors. The source concepts are likely represented as the abstraction of the common literal noun associations of metaphors, e.g., “half-baked ideas”, and “swallow that claim” are categorized as IDEAS ARE FOOD, where target IDEAS is the conceptualization of “ideas” and “claim”; source FOOD is the conceptualization of the common literal noun associations of “half-baked” and “swallow” (bread).

This is not the only pattern of source conceptualizations. Lakoff and Johnson also abstracted source concepts from the metaphoric words, e.g., they conceptualized “I demolished his argument” as ARGUMENT IS WAR, while WAR is the abstract concept of “demolished”, rather than its coherent literal associations (“demolish a war” is incoherent literally). We believe that the inconsistency is due to the subjective imageability of annotators because one can also abstract the concept as ARGUMENT IS BUILDING (“demolish a building” is coherent). The different imageabilities result in rich cognition about ARGUMENT, e.g., structure (BUILDING) and strategy (WAR). Here, we simplify the source conceptualization method by using common noun associations of metaphoric words to demonstrate our model with fewer variations.

A Wikipedia dump is parsed with Stanford Core NLP (Manning et al. 2014) to acquire common noun associations for metaphoric verbs and adjectives. We hypothesize that the frequent noun associations are likely literal, because according to relevant statistics, a third of sentences in typical corpora contain metaphorical expressions (Cameron 2003; Martin 2006; Steen et al. 2010; Shutova 2015). We use Wikipedia as the corpus for the frequency statistics, because scientific articles likely use literal expressions (Mao, Lin, and Guerin 2018). We count the frequency of each co-occurring word in Wikipedia, given  $w^s$  and the dependency of the word pair  $(w^s, w^t)$ . Top  $N$  (a hyperparameter) most frequent associations ( $a_m^s$ , where  $m \in [1, 2, \dots, N]$ ) of  $w^s$  are collected for inferring the source candidate concepts.

## Conceptualization

Next, we generate the source candidate concept ( $c_m^s$ ) for  $a_m^s$  and the target concept ( $c^t$ ) for  $w^t$ . Both  $a_m^s$  and  $w^t$  are nouns.

<sup>3</sup><https://dumps.wikimedia.org/enwiki/>

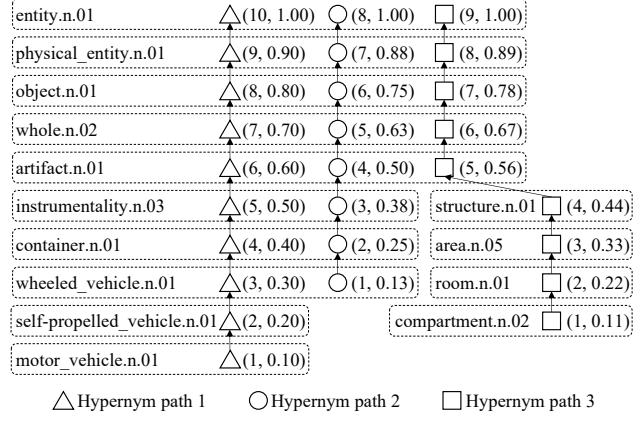


Figure 2: The rating scores of hypernyms of “car” in separate paths. The nodes in the same box denote the same synset in WordNet. The numbers in the parentheses besides a node denote the index of a node in a hypernym path (left), and the rating score in the path (right), respectively.

We defined a concept as an agent that represents the abstract meaning of a group of words. We use WordNet for conceptualization because it covers a wide range of concepts in the real world. Words are grouped (e.g., synonymy) and hierarchically structured (e.g., hypernym and hyponymy) by concepts in WordNet (Fellbaum 1998). A hypernym represents a broad meaning of specific words that fall under it, e.g., “furniture” is the hypernym of “bed” and “table”. Thus, hypernyms are eligible candidate concept agents.

There are three steps in the conceptualization procedure: (1) Obtain all WordNet hypernyms of a noun; (2) Rate the hypernyms with their abstract levels and the sense coverage significance; (3) Use the knee algorithm (Satopaa et al. 2011) and the hypernym rating scores to select an appropriate hypernym as a concept agent to represent the noun.

**(1)** By using the NLTK (Bird, Klein, and Loper 2009) Python package, we can obtain different paths from the node of a noun to the root node “entity” in WordNet. The nodes on the paths are hypernyms with different abstract levels (the higher the more abstract, the lower the more concrete). Different paths represent different senses of the noun. The paths will meet at a point and coincide if this point can summarize the meanings of all the hypernyms below. We define the set of all hypernyms of a noun as  $S_{noun}$ . We mean to select an appropriate hypernym from  $S_{noun}$ , representing the common sense of the noun as the concept agent, and keeping the selected hypernym concrete, e.g., “car” has three hypernym paths in Fig. 2: Path 1 denotes the sense of “a motor vehicle with four wheels”; Path 2 denotes “a wheeled vehicle adapted to the rails of railroad”; Path 3 denotes “the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant”.  $S_{car} = \{“motor\_vehicle”, “self-propelled\_vehicle”, “compartment”, “wheeled\_vehicle”, “room”, “container”, “area”, “instrumentality”, “structure”, “artifact”, “whole”, “object”, “physical\_entity”, “entity”\}$ .

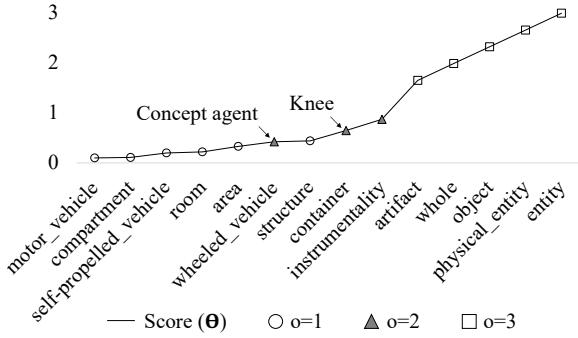


Figure 3: The rating score curve of hypernyms of “car” without smoothing and normalization.

(2) We rate the hypernyms to quantify their representations, according to their indices ( $i$ ) and the coverage of different hypernym paths ( $j$ ) of a noun. The index of the direct hypernym of a noun is 1 in each hypernym path. The index of the root node is the length of the path. The rating score ( $\Theta$ ) of a hypernym ( $hyper$ ) is given by

$$\Theta_{hyper} = \sum_j \frac{i_j^{hyper}}{len_j}, \quad (1)$$

where,  $len_j$  is the length of Path  $j$  that contains the hypernym, e.g., the lengths of Paths 1 and 2 are 10 and 8 in Fig. 2, where the indices of “container” are 4 and 2. Thus,  $\Theta_{container} = 4/10 + 2/8 = 0.65$ .

A larger  $\Theta$  means that the hypernym is more abstract, and covering more hypernym paths (senses) of a noun.

(3) We employ the knee algorithm (Satopaa et al. 2011) ( $K(\cdot)$ ) to balance the concreteness and sense coverage of a hypernym selection. A knee point is the point of the maximum curvature of a continuous function.  $K(\cdot)$  has embedded smooth and normalization functions to handle discrete data, e.g.,  $\Theta_{hyper}$ . Then, the knee point is  $knee = K(\Theta_{hyper})$ .

The knee point represents an approximate hypernym. A hypernym with a lower  $\Theta$  than the knee hypernym does not dramatically improve the concreteness level because its slope is flat on the curve (see an example later). In contrast, the hypernym with a lower  $\Theta$  may lose the significance in covering the common senses of a noun. In practice, we do not directly use the knee as the selected hypernym because the concreteness can be further improved without losing the sense coverage. If a hypernym with a lower rating score ( $\Theta_{hyper}$ ) covers the same number of hypernym paths as the knee point, it is as significant as the knee point in the sense coverage dimension, while it is more concrete than the knee point. We count the number ( $o_{hyper}$ ) of paths that include a given hypernym, and defining a set ( $D_{knee}$ ) where all  $o_{hyper}$  in the set are equal to  $o_{knee}$ . Then, the hypernym that is selected as the concept agent ( $c$ ) of a noun is given by

$$c = hyper* = \arg \min_{hyper \in D_{knee}} \Theta_{hyper}. \quad (2)$$

For example, in Fig. 3, the knee of “car” is “container”. It covers two hypernym paths ( $o = 2$ , see Fig. 2). The slope

before “container” is flat. “wheeled\_vehicle” ( $hyper*$ ; covering two paths; more concrete) is defined as the concept agent ( $c$ ) of “car”, because its rating score ( $\Theta$ ) is the lowest among all hypernyms that distribute in two hypernym paths.

## Metaphor Identification and Concept Generation

We have obtained the target concept agent ( $c^t$ ) from  $w^t$  and source candidate concept agents ( $C^s = \{c_1^s, \dots, c_m^s, \dots, c_N^s\}$ ) from the most frequent associations ( $a_m^s$ ) of  $w^s$ . We mean to generate an optimal source concept from  $C^s$  and identify the metaphoricity of the word pair ( $w^s, w^t$ ). To gain a broad concept spectrum, our model concept label vocabulary includes all WordNet nouns. Due to the absence of source gold labels, we employ a dynamic reward mechanism to push the learned concepts towards more accurate metaphoricity predictions. During training, given an input ( $w^s, w^t$ ), we first pair  $c_m^s$  and  $c^t$  based on a stochastic policy, where  $c_m^s \in C^s$ . The model learns  $c_m^s, c^t$ , and the metaphoricity label  $l$ , simultaneously, based on RoBERTa and MTL. Given input  $w^s$  and  $w^t$ , RoBERTa hidden states ( $h$ ) are given by

$$h^s, h^t = RoBERTa(< s >, w^s, w^t, < /s >), \quad (3)$$

where  $< s >$  and  $< /s >$  are special tokens, defined by RoBERTa;  $h^s$  is the RoBERTa hidden state, corresponding to  $w^s$ ;  $h^t$  corresponds to  $w^t$ . The predicted probability distributions ( $P$ ) of  $c_m^s, c^t$ , and  $l$  are given by

$$P(c_m^s) = softmax(W^s h^s + b^s), \quad (4)$$

$$P(c^t) = softmax(W^t h^t + b^t), \quad (5)$$

$$P(l) = softmax(W^l(h^s \oplus h^t) + b^l), \quad (6)$$

where  $W$  and  $b$  are learned parameters.  $\oplus$  is concatenation.

An accurate label prediction, higher  $P(\hat{l} = l)$  yields a higher reward to the losses of the concept predictions in E.q. 10, where the reward (the coefficient of the losses) is

$$\beta = \varphi P(\hat{l} = l)^2 + \gamma \mu. \quad (7)$$

$\varphi$  and  $\gamma$  are hyperparameters for balancing the losses of the label, source, and target concepts during training.  $\mu$  is

$$\begin{cases} \mu = sim(c_m^s, c^t), & \text{if literal;} \\ \mu = 1 - sim(c_m^s, c^t), & \text{otherwise.} \end{cases} \quad (8)$$

We mean to force the predicted source and target concepts more similar in literals, and more distinguishable in metaphors by  $\mu$ . We use Wu-Palmer similarity (Wu and Palmer 1994). The similarity  $sim(\cdot)$  is measured by

$$sim(c_m^s, c^t) = \frac{2 * N_{lcs \rightarrow root}}{N_{c_m^s \rightarrow lcs} + N_{c^t \rightarrow lcs} + 2 * N_{lcs \rightarrow root}}. \quad (9)$$

$lcs$  (least common subsumer) is the most specific ancestor concept, shared by two sub-concepts in WordNet (Pedersen et al. 2004).  $N_{lcs \rightarrow root}$  is the number of nodes on the path from the  $lcs$  of  $c_m^s$  and  $c^t$  to the root;  $N_{c_m^s \rightarrow lcs}$  is the number of nodes from  $c_m^s$  to  $lcs$ ;  $N_{c^t \rightarrow lcs}$  is the number of nodes from  $c^t$  to  $lcs$ .  $sim(c_m^s, c^t)$  is between 0 and 1.

We employ cross-entropy loss for the learning of each subtask ( $\mathcal{L}^l, \mathcal{L}^s, \mathcal{L}^t$ ). The overall loss ( $\mathcal{L}$ ) is given by

$$\mathcal{L} = \mathcal{L}^l + \beta \mathcal{L}^s + \beta \mathcal{L}^t. \quad (10)$$

Dataset	# WP	% M	% L	# UM
MOH	647	45.4	54.6	215
TSV-train	1455	47.8	52.2	28
TSV-dev	200	40.0	60.0	27
TSV-test	200	50.0	50.0	25
TSV-all	1855	47.2	52.7	28
GUT-train	6015	53.6	46.4	23
GUT-dev	859	53.4	46.6	23
GUT-test	1718	53.3	46.7	23
GUT-all	8592	53.6	46.4	23

Table 1: Dataset statistics. # WP denotes the number of word pairs. % M and % L are the percentage of metaphoric and literal word pairs among all word pairs, respectively. # UM is the number of unique metaphoric verbs or adjectives.

E.q.7-10 are applicable in the training procedure. In the inference procedure, the prediction of a source concept ( $c^{s*}$ ) is conditioned on the source candidate concept set ( $C^s$ ) by

$$c^{s*} = \arg \max_{\hat{c}_m^s \in C^s} P(\hat{c}_m^s). \quad (11)$$

The target concept prediction ( $c^t$ ) is given by  $w^t$  and the conceptualization algorithm in the previous subsection. The metaphoricity label prediction ( $l^*$ ) is given by

$$l^* = \arg \max_{\hat{l} \in \{\text{meta., lite.}\}} P(\hat{l}). \quad (12)$$

## Experiments and Results

### Baselines

**Multimodal** Shutova, Kiela, and Maillard (2016) proposed a word pair metaphor detection model with a multimodal learning method, incorporating word embeddings, visual embedding features, and a vector space similarity measure. **SSN** Rei et al. (2017) proposed a supervised similarity network, learning the cosine similarity patterns of metaphors with a gating mechanism and weighted cosine. The model incorporated different features, e.g., skip-gram and attribute-based vectors. We report the performance with the optimal feature setups (skip-gram for MOH and the fusion of two vectors for TSV).

### Datasets

**MOH** Shutova, Kiela, and Maillard (2016) developed a verb-noun pair dataset, parsed from the collection of Mohammad, Shutova, and Turney (2016). The dependent relationships between verb-noun pairs are either verb-subject or verb-direct object. We conduct 10-fold cross-validation with the MOH dataset for benchmarking.

**TSV** (Tsvetkov et al. 2014) is an adjective-noun pair dataset. We randomly sample 200 word pairs from the original training set as the development set.

**GUT** (Gutierrez et al. 2016) is an adjective-noun pair dataset. Compared with MOH and TSV, an adjective has more different noun associations in GUT. 72.5% words and 98.4% word pairs in GUT never appear in MOH and TSV. GUT is used for testing out-of-domain concept generations.

The detailed statistics are shown in Table 1.

Dataset	Model	P	R	F1	Acc
MOH	Multimodal	65	87	75	-
	SSN skip-gram	73.6	76.1	74.2	74.8
	Ours	72.5	79.3	75.6*	75.9
TSV	Multimodal	67	96	79	-
	SSN fusion	90.3	73.8	81.1	82.9
	Ours	89.4	84.0	86.6*	87.0

Table 2: Metaphor identification results, measured by F1 score. \* denotes the improvement is statistically significant, based on a two-tailed t-test ( $p < 0.01$ ).

### Setups

The batch size for training MOH is 256, while the batch size for training TSV, GUT, and the combination of MOH and TSV is 128<sup>4</sup>. We train the model with 40 epochs<sup>5</sup>. The reported testing results and 10-fold cross-validation results are based on the model that achieves the highest F1 score on the development sets. The model depends on Cuda 9.2 (NVIDIA, Vingelmann, and Fitzek 2020), Pytorch 1.7.1 (Paszke et al. 2019), and optimized with Adam optimizer (Kingma and Ba 2014) and a learning rate of 1e-5. We use RoBERTa-large as the encoder with a dropout rate of 0.3. In the common association acquisition procedure, we employ the 3 most frequent associations of a verb or an adjective in the Wikipedia dump.  $\varphi$  and  $\gamma$  in E.q. 7 are 0.05 and 0.005 respectively for balancing the losses between subtasks. If an input word is tokenized as Byte-Pair pieces (Sennrich, Hadlow, and Birch 2016) by RoBERTa, we use the first token as input to represent the original word.

### Metaphor Identification Evaluation

As seen in Table 2, our model exceeds the strongest baselines by 3.1% F1 (metaphors are positive labels) on average over the two datasets, where the TSV dataset (5.5%) yields higher gains than MOH (0.6%). This is because the size of TSV is larger than that of MOH. Our method learns more auxiliary conceptual information based on a larger dataset, thus yielding higher improvements on the main task.

There are three setups in our ablation analysis: (1) w/o MTL is a vanilla RoBERTa classification model without concept mapping MTL (the learnings of  $c_m^s$  and  $c^t$  in Fig. 1c are excluded); (2) w/o LB is the MTL model that excludes the loss balancing ( $\beta = 1$  in E.q. 10). (3) w/o DRW is the MTL model that excludes the dynamic reward mechanism, where we set up a fixed weight ( $\beta = 0.1$  in E.q. 10) to balance the subtask losses. We run the experiments on three datasets, MOH (10-fold), TSV development set, and TSV testing set, respectively. As seen in Table 3, our model (84.1%) achieves 6.4% average F1 gains, compared with the vanilla RoBERTa model (w/o MTL) (77.7%).

<sup>4</sup>We mean to use a larger batch size to improve the model performance (Liu et al. 2019). However, apart from MOH, 256 batch size runs out of memory on the other datasets, based on our employed model and GeForce GTX 1080 Ti GPU.

<sup>5</sup>This is because the model can converge in the metaphor detection task before 40 epochs

Setup	MOH		TSV-dev		TSV-test		Avg F1
	F1	Acc	F1	Acc	F1	Acc	
w/o MTL	73.6	74.4	85.5	89.0	74.1	78.0	77.7
w/o LB	70.8	72.4	80.5	84.0	77.5	80.5	76.3
w/o DRW	73.9	75.4	88.1	90.1	79.6	81.2	80.5
Ours	<b>75.6</b>	75.9	<b>90.2</b>	92.0	<b>86.6</b>	87.0	<b>84.1</b>

Table 3: Ablation analysis.

This shows that the auxiliary concept learning task and the dynamic reward mechanism are supportive of metaphor identification. This is because the model prediction simulates the decision mechanism of human metaphor detection, predicting the metaphoricity label, based on the source and target hidden states ( $h^s$  and  $h^t$  in Fig. 1c). Conversely, without the dynamic reward mechanism and loss balancing, the model (w/o LB vs. ours) shows a sharp drop in the average F1 score (-7.8%). This is because the learning of concepts is much more difficult than the learning of metaphoricity labels. Without balancing the losses, the model may be ruined by the concept learning. w/o DRW surpasses w/o MTL by 2.8% F1 on average. This shows the effectiveness of introducing the concept learning auxiliary task and balancing losses between subtasks. Finally, the full model yields 3.6% gains, compared with w/o DRW, which signifies the utility of the proposed dynamic reward mechanism.

### Concept Mapping Automatic Evaluation

We test the generated source and target concept agents as features for word pair metaphor detection. A different dataset is employed to evaluate out-of-domain concepts. The testing model is based on a typical RoBERTa-large sequence classification model. The input is two generated concepts of a given word pair. The object is to classify the metaphoricity, based on concepts. To the best of our knowledge, we are unable to find an appropriate external baseline, functioned with source and target concept generations for benchmarking.

We use the conceptualization algorithm and the pre-trained model (a concept generator) that was trained on the combination of the MOH and TSV training sets to generate concepts for word pairs in the GUT. We employ an early stop. The model stops training when the accuracy of the metaphor identification task reaches 100% on the training set, because the rate of reward starts to be less distinctive. The baseline model (rand-candidate) is based on generated target concepts and randomly selected source concepts from the source candidate sets ( $C^s$  in Fig. 1b). We also include the baseline model (rand-concept) whose input is randomly selected WordNet hypernyms of target words and the frequent associations of source words. The knee point selection algorithm is excluded in rand-concept. Then, the concept agent of “car”, e.g., can be any hypernym in Fig. 3. All the above-mentioned methods are ultimately compared to the model (original) that is trained with the original word pairs. We maintain the concept consistency of the rand-candidate and rand-concept inputs (a word is represented as a fixed concept). We report the RoBERTa model performance on the testing set with different inputs in Table 4.

Setup	P	R	F1	Acc	$\Delta$ F1
Original	98.0	97.9	98.0	97.9	-
Rand-concept	79.8	85.2	82.4	80.6	-15.6
Rand-candidate	90.6	93.1	91.8	91.2	-6.1
Ours	91.3	93.7	92.5	91.9	-5.5

Table 4: Automatic evaluation for concept mappings, based on GUT testing dataset.

As seen in Table 4, the benchmark model (original) achieves a very accurate result (98.0% F1) in metaphor identification. This is because an adjective has a large number of different noun associations to learn the metaphoricity (see Table 1). Our generated concept inputs yield a marginal loss ( $\Delta = -5.5\%$  F1), compared with the original word pairs, due to the errors of the concept generator. Using concept training and testing sets from different domains improves the difficulty of the task. Nevertheless, the classifier with our generated concepts still yields acceptable predictions (92.5% F1). The gap between the randomly selected candidate concept (rand-candidate) and the original word pair input increases slightly ( $\Delta = -6.1\%$  F1). However, it is still within the acceptable range. This shows that the statistical knee algorithm-based target and source candidate conceptualizations (see Figures 1a and 1b) are qualified features in machine learning. The MTL procedure (Fig. 1c) further improves the quality of the generated concepts. Without the MTL and conceptualization algorithms, we observe a sharp drop in the rand-concept baseline ( $\Delta = -15.6\%$  F1). Even though the random concepts are also the hypernyms of the input word pairs, the MTL and conceptualization algorithms can generate more robust concepts, yielding 10.1% gains.

### Hyperparameter Analysis

We evaluate the number of the most frequent associations that was manually defined in the section of common association acquisition. We test the hyperparameter by using the MOH 10-fold cross-validation and TSV development sets (the metaphor detection task), and the GUT development set (the concept feature evaluation task<sup>6</sup>). As seen in Table 5, the top 3 most frequent associations of the source word tend to yield the optimal results. For other values ( $N < 3$ ), the performance decreases, because the candidate source concepts cannot be effectively covered by the source word associations; Alternatively ( $N > 3$ ), more association words may introduce additional noise for the source concept learning.

### Concept Mapping Human Evaluation

Next, we invite three participants to evaluate the generated concept mappings of metaphoric word pairs. The participants are psycho-linguistic research students whose mother language is English. We develop a test set by randomly selecting 100 metaphoric word pairs from MOH, TSV, and GUT testing sets, repetitively (totally, 300).

<sup>6</sup>We train different concept generators (see the section of concept mapping automatic evaluation) with different numbers of the most frequent associations to align to the downstream evaluation.

Task	# freq. assoc.				
	1	2	3	4	5
MOH	71.3	72.8	75.6	74.9	72.1
TSV	82.2	89.2	90.2	88.8	88.7
GUT	91.3	91.6	91.6	91.2	90.7
Avg	81.6	84.5	85.8	84.8	83.8

Table 5: Hyperparameter analysis, measured by F1. # freq. assoc. denotes the number of the most frequent associations.

The baseline (rand-candidate) yields randomly selected concepts from the source candidate set ( $C^s$ ) for benchmarking. The target concept is the same between the two models. We evaluate the accuracy of a concept mapping with two questions. Given a word pair (a metaphoric verb or adjective, associated with a literal noun), their dependency, and a source target concept pair, the questions are: Q1. Whether the noun is conceptually mapped to the source concept? Q2. Whether the basic meaning of the noun belongs to the target concept? Annotators are encouraged to answer the questions with contexts, e.g., given “son *drift*” in subject-verb dependency, whether “son” is conceptually mapped to the source concept VESSEL? Whether the basic meaning of “son” belongs to the target concept MALE\_OFFSET? An example sentence for your reference is “my son *drifted* around for years in California before going to law school” (Fellbaum 1998). The final annotation of each question is agreed upon by at least two annotators (more than half). We use Cohen (1960)’s kappa ( $\kappa$ ) as an agreement measure, where  $\kappa_{source} = 0.87$  and  $\kappa_{target} = 0.89$ . The performance is measured by accuracy (the number of correct concept generations above the total number of cases). An accurate concept mapping means that both the source and target concepts (Q1 and Q2) are correct. As seen in Table 6, both our method and the rand-candidate baseline yield positive results in the source (70.7% vs. 67.3%) and target (87.3%) concept generation, and concept mapping (63.7% vs. 59.3%) evaluation tasks overall. It supports the automatic evaluation results in Table 4. We observe that the accuracy of source concept generations is often lower than the target, because the source concepts are indirectly inferred from the frequent associations (see the section of common association acquisition). However, the accurate target concept generations (above 84% across the three datasets) demonstrate the effectiveness of our conceptualization method.

## Case Study

Some concept mapping examples can be viewed in Table 7. The model generates the concept mapping DOCUMENT IS WAY for “steamroller bill” (M1). It explains the metaphoricity of *steamroller* because “bill” (DOCUMENT) maps to a different concept domain (WAY) in a context that contains the word pair, such as “the Senator *steamrolled* the bill to defeat” (Fellbaum 1998). Our method shows the implicit imageabilities of “steamroller” and “bill” that are not directly presented in the sentence. Another example is that “steep discount” (T5, DECREASE IS GEOLOGICAL\_FORMATION) is metaphoric because DECREASE (target) and GEOLOGI-

Dataset	Method	Source (Q1)	Target (Q2)	Mapping (Q1 & Q2)
MOH	Rand-candidate	62.0	86.0	56.0
	Ours	67.0	86.0	62.0
TSV	Rand-candidate	66.0	92.0	63.0
	Ours	69.0	92.0	67.0
GUT	Rand-candidate	74.0	84.0	59.0
	Ours	76.0	84.0	62.0
All	Rand-candidate	67.3	87.3	59.3
	Ours	70.7	87.3	63.7

Table 6: Human evaluation for concept generations and concept mappings, measured by accuracy.

Word pair	DR	Target	Source
M1 <i>steamroller</i> bill	vo	DOCUMENT	WAY
M2 son <i>drift</i>	sv	MALE_OFFSET	VESSEL
M3 wine <i>breathe</i>	sv	ALCOHOL	ADULT
M4 story <i>lend</i>	sv	FICTION	ADULT
M5 government <i>bow</i>	sv	POLITY	ADULT
T1 <i>blind</i> alley	an	STREET	ADULT
T2 <i>raw</i> emotion	an	FEELING	ARTIFACT
T3 <i>weak</i> password	an	POSITIVE_ID	ARTIFACT
T4 <i>rough</i> draft	an	WRITING	ARTIFACT
T5 <i>steep</i> discount	an	DECREASE	GEO_FORM
G1 <i>bitter</i> night	an	TIME_PERIOD	SENSATION
G2 <i>sour</i> trade	an	TRANSACTION	SENSATION
G3 <i>clear</i> definition	an	EXPLANATION	MATERIAL
G4 <i>warm</i> gratitude	an	FEELING	MATERIAL
G5 <i>clean</i> datum	an	INFORMATION	MATERIAL

Table 7: Case study. M, T, and G are MOH, TSV, and GUT datasets. DR denotes dependency relationship, where sv, vo, and an are subjective-verb, verb-direct object, and adjective-noun dependencies, respectively.

CAL\_FORMATION (source) are from two different concept domains. Additionally, as seen in the source column in Table 7, the word pairs are categorized by their abstract source concepts, e.g., “wine *breathe*”, “story *lend*”, “government *bow*” and “*blind* *alley*” can be categorized by the ADULT source concept. Both “*bitter* *night*” and “*sour* *trade*” map to the same SENSATION source concept.

## Conclusion

In this work, we propose a word pair-level metaphor identification method. The method can identify the metaphoricity and generate source and target concepts in natural language for a dependent word pair input. Such an approach allows the metaphor identification output to be explainable in Conceptual Metaphor Theory. We demonstrate that the model yields better performance than previous word pair-level metaphor identification baselines.

In addition, we show the utility of the generated concepts with automatic and human evaluation tasks and a dataset that contains out-of-domain concepts. The generated source and target concepts are categorized and structured in line with WordNet. Thus, our model can be a useful tool for supporting conceptual metaphor corpus study in future work.

## Acknowledgments

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

## References

- Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.
- Birke, J.; and Sarkar, A. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 329–336.
- Cameron, L. 2003. *Metaphor in Educational Discourse*. A&C Black.
- Choi, M.; Lee, S.; Choi, E.; Park, H.; Lee, J.; Lee, D.; and Lee, J. 2021. 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*, 1763–1773.
- Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1): 37–46.
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Gao, G.; Choi, E.; Choi, Y.; and Zettlemoyer, L. 2018. Neural Metaphor Detection in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 607–613.
- Gutierrez, E. D.; Shutova, E.; Marghetis, T.; and Bergen, B. 2016. Literal and metaphorical senses in compositional distributional semantic models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 183–193.
- Heintz, I.; Gabbard, R.; Srinivasan, M.; Barner, D.; Black, D. S.; Freedman, M.; and Weischedel, R. 2013. Automatic extraction of linguistic metaphor with LDA topic modeling. In *Proceedings of the 1st Workshop on Metaphor in NLP*, 58–66.
- Howard, N.; and Cambria, E. 2013. Intention awareness: Improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences*, 3(9).
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lakoff, G.; Espenson, J.; and Schwartz, A. 1991. *Master Metaphor List*. University of California at Berkeley, 2nd edition.
- Lakoff, G.; and Johnson, M. 1980. *Metaphors We Live by*. University of Chicago press.
- Leong, C. W.; Klebanov, B. B.; Hamill, C.; Stemle, E.; Ubale, R.; and Chen, X. 2020. A report on the 2020 VUA and TOEFL metaphor detection shared task. In *Proceedings of the 2nd Workshop on Figurative Language Processing*, 18–29.
- Leong, C. W. B.; Klebanov, B. B.; and Shutova, E. 2018. A report on the 2018 VUA metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, 56–66.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 55–60.
- Mao, R.; and Li, X. 2021. 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*, 13534–13542.
- Mao, R.; Lin, C.; and Guerin, F. 2018. Word embedding and WordNet based metaphor identification and interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1222–1231.
- Mao, R.; Lin, C.; and Guerin, F. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3888–3898.
- Martin, J. H. 2006. A corpus-based analysis of context effects on metaphor comprehension. *Trends in Linguistics Studies and Monographs*, 171: 214.
- Mason, Z. J. 2004. CorMet: a computational, corpus-based conventional metaphor extraction system. *Computational linguistics*, 30(1): 23–44.
- Mohammad, S.; Shutova, E.; and Turney, P. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, 23–33.
- NVIDIA; Vingelmann, P.; and Fitzek, F. H. 2020. CUDA, release: 9.2.148.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An imperative style, high-performance deep learning library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.
- Pedersen, T.; Patwardhan, S.; Michelizzi, J.; et al. 2004. WordNet: Similarity-measuring the relatedness of concepts. In *AAAI*, volume 4, 25–29.
- Rei, M.; Bulat, L.; Kiela, D.; and Shutova, E. 2017. Grasping the finer point: A supervised similarity network for

metaphor detection. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1537–1546.

Satopaa, V.; Albrecht, J.; Irwin, D.; and Raghavan, B. 2011. Finding a “needle” in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, 166–171. IEEE.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725.

Shutova, E. 2015. Design and evaluation of metaphor processing systems. *Computational Linguistics*, 41(4): 579–623.

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

Shutova, E.; and Simone, T. 2010. Metaphor corpus annotated for source-target domain mappings. In *Proceedings of the International Conference on Language Resources and Evaluation*, 3255–3261.

Shutova, E.; Sun, L.; Gutiérrez, E. D.; Lichtenstein, P.; and Narayanan, S. 2017. Multilingual metaphor processing: Experiments with semi-supervised and unsupervised learning. *Computational Linguistics*, 43(1): 71–123.

Shutova, E.; Sun, L.; and Korhonen, A. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 1002–1010.

Steen, G. J.; Dorst, A. G.; Herrmann, J. B.; Kaal, A.; Krennmayr, T.; and Pasma, T. 2010. *A Method for Linguistic Metaphor Identification: from MIP to MIPVU*, volume 14. John Benjamins Publishing.

Stowe, K.; Moeller, S.; Michaelis, L.; and Palmer, M. 2019. Linguistic analysis improves neural metaphor detection. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 362–371.

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

Tsvetkov, Y.; Boytsov, L.; Gershman, A.; Nyberg, E.; and Dyer, C. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 248–258.

Wu, Z.; and Palmer, M. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, 133–138.

Zayed, O.; McCrae, J. P.; and Buitelaar, P. 2020. Contextual Modulation for Relation-Level Metaphor Identification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 388–406.