Can ChatGPT's Performance be Improved on Verb Metaphors Detection Tasks? Bootstrapping and Combining Tacit Knowledge

Anonymous ACL submission

Abstract

Metaphors detection, as an important task in the field of NLP, has been receiving sustained academic attention in recent years. Current researches focus supervised metaphors detection systems, which usually require large-scale, high-quality labeled data support. The emerge of large language models (e.g., ChatGPT) has made many NLP tasks (e.g., automatic summarization and dialogue systems) a qualitative leap. However, it is worth noting that the use of ChatGPT for unsupervised metaphors detection is often challenged with less-than-expected performance. Therefore, the aim of our work is to explore how to bootstrap and combine ChatGPT by detecting the most prevalent verb metaphors among metaphors. Our approach first utilizes ChatGPT to obtain literal collocations of target verbs and subject-object pairs of verbs in the text to be detected. Subsequently, these literal collocations and subject-object pairs are mapped to the same set of topics, and finally the verb metaphors are detected through the analysis of entailment relations. The experimental results show that our method achieves the best performance on the unsupervised verb metaphors detection task compared to existing unsupervised methods or direct prediction using ChatGPT.

1 Introduction

Metaphors are essentially mapping relationships between two different domains (Hesse, 1965; Lakoff and Johnson, 2008). According to the conceptual metaphor theory (Lakoff and Johnson, 2008), linguistic metaphors derive from underlying conceptual metaphors that map a source concept to another, more abstract, target concept.

Metaphors detection aims at modeling non-literal expressions (e.g., metaphors and metonymy) and generating corresponding metaphorical annotations. It is beneficial to many NLP tasks, e.g., information extraction (Tsvetkov et al., 2013), sentiment analysis (Cambria et al., 2017), and machine translation (Babieno et al., 2022).

In previous researches, most metaphors detection methods have primarily used supervised approaches (Song et al., 2021; Zhang and Liu, 2023). Although these models achieve excellent performance on test sets, they rely on well-labeled datasets and evidently suffer from low generalization performance when shifting to different domains (Wang et al., 2023). In addition, high-quality labeling data is time-consuming and expensive, especially for metaphor samples, which require more complex and difficult expert knowledge to perform data labeling.

To cope with the above problems, researchers have explored the unsupervised domain. Heintz et al. (2013) constructed a topic list using latent derechter allocation (LDA) (Blei et al., 2003). Shutova and Sun (2013) constructed a clustering map. Gandy et al. (2013) and Pramanick and Mitra (2018) introduced lexical abstraction to study copular verbs metaphor and real verbs metaphor, respectively. However, these methods usually require complex hand-coding rules. To simplify the methods, Mao et al. (2018) and Shutova et al. (2016) used cosine distance to determine whether subject-verb or verb-object pairs belong to the same conceptual domain. Nevertheless, these methods still rely on partially manually labeled datasets.

With the development of large language models (LLMs), and in particular ChatGPT's excellent performance on zero-shot or few-shot NLP tasks (Yoo et al., 2021; Meng et al., 2022), we are inspired to consider utilizing world knowledge of ChatGPT to augment a metaphors detection system. Given that verb metaphors occupy the broadest class of metaphors (Shutova and Teufel, 2010), many supervised (Song et al., 2021) and unsupervised methods (Mao et al., 2018; Shutova et al., 2016) focus on verbs, our work likewise concentrates on the verb part. We propose an unsupervised verb metaphors detection method based on ChatGPT. First, we build a verb list that records the literal meaning collocation of each verb. Then, we introduce topical features that map the subject and object of the target verb to one or more topical categories. Next, we analyze the subjects and objects of the verbs to be detected in the input text and map them to topical categories as well. Finally, we detect verb metaphors through the analysis of entailment relations. We test our model on the VUAverb, MOH-X, and TroFi datasets, and the results show that by bootstrapping and integrating the implicit knowledge of ChatGPT, it can effectively improve performance on the verb metaphors detection task.

In summary, the main contributions of our work are summarized as follows:

1. We are the first to introduce ChatGPT to the verb metaphors detection task and do not need to rely on tedious hand-coding rules or manually labeled data.

2. We use ChatGPT to generate a verb list that provides reference information about the literal collocation of each verb. We introduced topical features to map the target vocabulary to more general concepts.

3. We compare our method with previous unsupervised methods and direct use of the ChatGPT method. Experiments demonstrate that our method achieves the best performance on three datasets, VUAverb, TroFi and MOH-X.

4. We compare the proposed method with zero-shot and few-shot sample generation methods. These methods utilize ChatGPT to generate or introduce examples to generate metaphorical samples, which are subsequently fine-tuned using a pre-trained model. Our approach similarly achieves the best performance.

2 Related Work

To minimize the reliance on labeled data, researchers have explored a lot on unsupervised methods. Karov and Edelman (1998) used a word sense disambiguation (WSD) algorithm to cluster sentences with target words, and then made metaphor predictions based on the principle of distance between literal meanings of words. Shutova and Sun (2013) also drew on the idea of clustering, and it used the Gigaword corpus (Graff et al., 2003) with noun-related of verb-noun combinations (grammatical features) to cluster the 2000 common nouns of the BNC. In this approach, the words to be detected acquire knowledge information at a certain layer in the clustering map, i.e., the nouns at that layer are non-metaphorically related to the words to be detected.

Mao et al. (2018) presented an approximately unsupervised metaphors detection system. The system selects the best alternative to the target word by considering superlatives and synonyms in the context. When the cosine distance between the best alternative and the target word is greater than a specific threshold, it is detectd as a literal meaning. In addition, other studies (Shutova et al., 2016; Pramanick and Mitra, 2018) have considered the cosine distance, although Pramanick and Mitra (2018) did not use a priori labeled data to set the threshold, instead it adopted a feature construction approach using clustering for metaphorical judgments.

Some studies (Turney et al., 2011; Gandy et al., 2013) explored the relationship between the abstraction degree of focus words and the expression of language metaphors. Turney et al. (2011) used the abstraction degrees of nouns, proper nouns, verbs and adverbs were first calculated, and then logistic regression to learn high-dimensional metaphoric features. Gandy et al. (2013) used WordNet to generate n common collocations of the words to be detected and sorted these collocations according to the abstraction level. A metaphorical relationship word is detectd as a metaphor if it is not between the first k most concrete words. This idea is also reflected in the study of Krishnakumaran and Zhu (2007), which investigated three metaphorical relations, Subject-be-Object, Verb-Object and Adjective-Noun, and identified metaphors by determining whether the two focal words have a hyponymy relation.

Although the above methods have been effective to a certain extent, there are still problems such as complex parsing of metaphorical relationships, cumbersome construction of hand-coded knowledge, or reliance on manually labeled data. To overcome these challenges, we attempt to introduce generative language modeling into the metaphors detection task. The main function of generative language models is to generate natural language text, which can be used for conversing with humans or performing text generation tasks. In previous research, Wachowiak and Gromann (2023) used

GPT-3 for supervised metaphor generation. The study first provided input text and target domain information, and then utilized GPT-3 to predict source domain information. This is a good attempt, but still relies on labeled data. The difference is that our study focuses on unsupervised method to acquire implicit knowledge of ChatGPT through bootstrapping and integration. Our approach achieves significant performance gains in the unsupervised metaphor detection task.

3 Method

We divide the proposed method into three parts: definition of verb metaphors, topic mapping and verb list.

3.1 Defining Verb Metaphors

Our study on verb metaphors is based on the theory of selectional preference violation (SPV) (Wilks et al., 2013). As an important concept in linguistics, SPV reflects the relatedness and semantic compatibility between lexical units. For example, in the phrase "kill time", the verb "kill" is originally preferred to describe the behavior of animate objects, but here it modifies the inanimate "time", so there is a case of selectional preference violation.

Previous studies (Shutova et al., 2012, 2016) usually categorized verb-metaphor relations into two main types, i.e., Subject-Verb (SV) pair and Verb-Object (VO) pair. For example, in the sentence "He planted good ideas in their minds.", "ideas" is the object of the verb, and the verb "planted" forms a VO pair with "ideas", while the subject of the target verb "planted" is "he", which forms an SV pair. To capture the metaphorical relations of verb pair more comprehensively, we consider both SV pair and VO pair. We consider the target verb to be non-metaphorical only if both sub-relations exhibit literal meaning relations. Other studies (Krishnakumaran and Zhu, 2007; Gandy et al., 2013) have also introduced Subject-be-Object (SbeO) relations. For example, in the sentence "Her love is a warm blanket on a cold night.", "love" is metaphorized as a warm blanket. In this structure, the verb "is" connects two focus words, "love" and "blanket". However, it should be noted that "is" as an auxiliary verb does not have an independent lexical meaning by itself, and it needs to be combined with other verbs. Therefore, when judging the metaphor of SbeO structures, it is necessary to consider whether there is an entailment relationship between the subject or object. This is relatively similar to the Adjective-Noun (AN) relationship (Pramanick and Mitra, 2018), e.g., the SbeO structure "love is warm" with the AN structure "warm love". Therefore, we categorize SbeO relations in the same category as AN pairs instead of including them in verb metaphors.

3.2 Topic Mapping

Metaphorical relationships originated from conceptual mappings in different domains (Lakoff and Johnson, 2008). Inspired by it, we introduce the concept of topic, which can be viewed as broader and abstract concepts to correspond to domains in metaphors. Consider an example of a verb metaphors using the Oxford topics, the verb "guzzle" is often used with the subjects "baby" and the objects "milk". However, in the sentence "The car guzzled down the gasoline.", the subject and object of the target verb "guzzled" are "car" and "gasoline", respectively. This leads to the selectional preference violation. In addition, since "bus" or "taxi" belongs to the same topic "Transport by car or lorry" as "car". Therefore, replacing the subject of the above example sentence with "bus" or "taxi" also constitutes a metaphorical expression.

Subject(Topic)	Object(Topic)
person (people)	Food or meals (Cooking and eating)
Children (Life stages)	Snacks (Cooking and eating)
Adults (Life stages)	Meat (Food)
diners (Cooking and eating)	Vegetables (Food)

Table 1: The subject and object of the verb "eat" are literally paired, with the corresponding Oxford topics category indicated in parentheses.

We introduce three kinds of topics, namely Oxford topics, WordNet topics, and LDA topics. These three topic categories are set up in line with both the SPV (Wilks et al., 2013) and the abstractness principle (Turney et al., 2011; Gandy et al., 2013). The principle of abstraction holds that focus words under the same topic usually have similar or close levels of abstraction. For example, in the example in the Oxford topics, "Anger", "Fear" and

"Happiness" all belong to the "People-Feelings" topical category, and these words have similar levels of abstraction. However, it is important to note that, since a single word may have more than one denotation, the word may correspond to more than one different Oxford topics. The LDA topics (Heintz et al., 2013) were derived from a category list containing 60 topics. The method first used the LDA (Blei et al., 2003) model to capture a variety of candidate topics from WiKipedia. Then, based on the metaphorical information contained in the input corpus, the topics with high relevance to metaphorical relations were selected as the final metaphorical topics, and they were summarized into 60 different topic categories. The constructed topics would be categorized according to the order of similarity in WordNet from high to low for the central words.

Similar to the infix relation (Krishnakumaran and Zhu, 2007), we introduce the set of superlatives and synonyms in WordNet (Kilgarriff, 2000) as a third topic (WordNet topics). In WordNet, superordinates are defined as semantically more general or abstract words, while synonyms denote words with similar or identical meanings that can provide complementary information. Since both superlatives and synonyms are considered, each central word in a WordNet topics contains all synonyms and superlatives compared to LDA topics that select one or more topics by similarity.

3.3 Construction of Verb Lists

Supervised models tend to exhibit a sharp drop in performance in new domains (Wang et al., 2023), revealing the problem of domain bias. Domain bias indicates that the metaphorical dataset is significantly different from the actual application environment. As a result, models trained on traditional datasets may have difficulty adapting to the metaphor usage context of real application domains.

To address this challenge, we construct a verb literal meaning collocation list that requires no additional training and can be applied to detect samples with different distributions. The verb list requires no additional training and can be applied to detect samples with different distributions. For the construction of verb list, we used GPT-3.5 Turbo (hereafter Turbo) to generate literal or non-metaphorical collocations of verbs. Turbo is a lightweight text generation model developed by OpenAI that can be adapted to a variety of use cases through fine-tuning. First, we use the Turbo to generate subject and object collocations for the target verbs (See Appendix §11.1 for details of prompt design). Then, SV and VO pairs are extracted respectively by regular expressions and stored as a list. Noting that each target verb corresponds to two lists (i.e., the subject list and the object list), which do not correspond to each other. Next, we map the subject and object contents of the lists to one or more topics (see §3.2 for details), and the same topics for the same verb will be merged. Table 1 shows the Oxford topics information for the verb "eat". In the list, both "Children" and "Adult" belong to the topical category "Life stages", so they are merged into the same category. Similarly, the object content of "Food and meals", "Snacks", "Meat" and "Vegetables" are categorized respectively.

3.4 Method Implementation Details

The details of the algorithm can be found in Algorithm 1. First, we build a list of containing verbs D as described in §3.3. This verb list is in the form of a dictionary, where each particular verb is used as an indexing keyword, and the corresponding subject or object is stored in the form of a list, labeled as S_w and O_w, respectively. To perform metaphors detection, the input text needs to be processed first. Similar to the manipulation of verb lists, we will extract the subject and object in each input text. In previous studies, researchers (Wilks et al., 2013; Shutova et al., 2016; Gandy et al., 2013) usually used the stanford dependency parser to extract SV and VO pairs of metaphorical relations, while Krishnakumaran and Zhu (2007) employed PCFG (Klein and Manning, 2003) for grammatical parsing. However, these approaches usually require the specification of complex rules to take into account complex grammatical structures such as inversions, implied subjects or objects, and subordinate clauses. Therefore, we use the ChatGPT3.5-Turbo to generate the subject-verb-object (see Appendix §11.2 for details of the prompt design). We then use regular expressions to parse the results generated by Turbo and store them as a list. If the generated SV or VO pair contain pronouns or named entities, we first obtain their basic meanings in the Oxford dictionary. For example, "it" corresponds to "used to refer to an animal or a thing that has already been mentioned or that is being talked about now". In this case, we usually choose the

Algorithm 1 Metaphors Detection

Require: D: Dictionary of verb forms
Require: S_w: List of literal or non-metaphorical subject topics for each target verb
Require: O_w: List of literal or non-metaphorical object topics for each target verb
Require: N: Input corpus containing sentences with target verbs
Require: w_n: Target verb in sentence n
Require: i_n: Index of the target verb in sentence n

for n in N **do**
  $S_{w_n} \leftarrow D[w_n][0]$ ▷ Retrieve subject topics
  $O_{w_n} \leftarrow D[w_n][1]$ ▷ Retrieve object topics
  Extract the **subject** and **object** from the sentence at index i_n.
  subj_nouns $\leftarrow$ get_top_k_noun(subject)
  obj_nouns $\leftarrow$ get_top_k_noun(object)
  subj_topics $\leftarrow$ get_topics_from_oxford(subj_nouns)
  obj_topics $\leftarrow$ get_topics_from_oxford(obj_nouns)
  if_sub_literal $\leftarrow$ subj_topics $\in S_{w_n}$ ▷ Is subject literal?
  if_ob_literal $\leftarrow$ obj_topics $\in O_{w_n}$ ▷ Is object literal?
  if $\neg$(if_sub_literal $\wedge$ if_ob_literal) **then**
    if_metaphor $\leftarrow$ **True** ▷ Metaphor detected
  else
    if_metaphor $\leftarrow$ **False** ▷ No metaphor
  end if
end for

first 3 nouns (if they exist) as the center words of "it", such as "animal" and "thing".

Since the subjects and objects in the SV or VO pair output by the model are usually presented as phrases, we will select the first k nouns in the phrases as the center words of the subjects or objects and notate them as "subj_nouns" and "obj_nouns", respectively. Then, depending on the lexical meaning of these center words, we map them to one or more topics, denoted as "subj_topics" and "obj_topics", respectively. For example, in the sentence "He was detained on June 23, and for two weeks he was regularly assaulted by South African police", the subject of the sentence is "South African police". We extract the first k nouns as the center word, i.e., "police" ($k = 1$). According to the lexical meaning, we map "police" to the Oxford topic "Law and justice". Finally, we make metaphorical judgments based on the relationship between the parsed topics and the reference topics in the verb list.

4 Experiments

4.1 Test Datasets

VUAverb. The vu amsterdam metaphor corpus (Steen et al., 2010) metaphorically annotates each lexical unit from a subset of the british national corpus (Edition et al.). The annotation was done with high inter-annotator agreement and a Kappa value greater than 0.8. The VUAverb is a verb part extracted from the VUA. We used the test set reported in the metaphors detection shared task (Leong et al., 2018, 2020) in our experiments. The test set contains 5,873 samples.

TroFi. The TroFi dataset (Birke and Sarkar, 2006) is derived from the wall street journal corpus (Charniak et al., 2000). In the original TroFi dataset, each sample is annotated with one of three labels: L (literal), N (non-literal), or U (unannotated). We used the (Leong et al., 2018, 2020) version of the TroFi dataset, which includes literal and metaphorical usage of 50 English verbs, totaling 3,717 samples, as examples of verb metaphors.

MOH-X. The MOH dataset (Mohammad et al., 2016)was labeled metaphorically through a crowdsourcing platform for sentences. To ensure the quality of the annotation of the dataset, Mohammad et al. (2016) adopted the principle of 70% annotation consistency. We considered the subset of verbs in the MOH dataset, MOH-X (Shutova et al., 2016). This subset excludes instances with pronouns or subordinate subjects or objects. The dataset ultimately contains 647 pairs of verb-noun

Models	VUAverb				TroFi				MOX-H			
	Acc.	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**
Concrete-Abstract	44.7	31.3	66.3	42.5	51.6	46.3	69.9	55.7	55.5	53.3	67.0	59.3
WORDCOS	38.3	31.5	88.0	46.4	46.2	44.2	89.8	59.2	46.4	47.4	90.7	62.3
SIM-CBOW	38.0	31.6	89.5	46.7	44.9	43.8	93.9	59.7	48.6	48.6	94.6	64.2
GPT-3.5 Turbo	65.2	33.4	14.8	20.5	58.7	64.2	11.4	19.3	60.1	91.3	20.0	32.8
Ours (llama2)	30.6	30.1	97.8	46.1	43.9	43.6	98.6	**60.5**	50.1	49.4	97.5	65.6
Ours (turbo)	45.4	34.6	90.3	**50.0**	45.8	44.2	93.7	60.1	61.2	56.1	93.3	**70.1**

Table 2: Comparison with the baseline models. Both SIM-CBOW and WORDCOS are encoded using CBOW and word distances are computed with cosine similarity. Concrete-Abstract introduces lexical specificity. Our approach uses llama2 or GPT-3.5 Turbo to construct verb list and then adopts the Oxford Dictionary as a topic mapping tool.

Models	VUAverb				TroFi				MOX-H			
	Acc.	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**
DG (zero-shot)	61.1	78.5	31.9	45.3	59.7	57.9	27.1	37.0	71.1	56.2	18.9	28.3
EPE (few-shot)	69.7	49.8	56.7	**53.0**	57.1	50.6	68.8	58.3	60.1	62.5	53.1	57.4
Ours (turbo)	45.4	34.6	90.3	50.0	45.8	44.2	93.7	**60.1**	61.2	56.1	93.3	**70.1**

Table 3: Comparison with the sample generation methods. As with our approach, both the Direct Generation (DG) and Example Prompt Enhancement (EPE) methods use ChatGPT 3.5 Turbo. EPE gives an example of manual annotation for both the given verb and the label (metaphorical or literal).

combinations of which 316 pairs are metaphorical and 331 pairs are literal.

4.2 Experimental Setup

Experiment 1. Experiment 1 demonstrates the performance of our unsupervised approach. We chose three baseline models (Mao et al., 2018; Shutova et al., 2016; Turney et al., 2011) for the previous unsupervised methods. For the LLMs, we used both LLaMA and ChatGPT-3.5 Turbo for constructing verb list. Finally, we will use GPT-3.5 Turbo directly as a control.

In the unsupervised approach, Mao et al. (2018) introduced synonyms and superlatives in WordNet, calculated the best match by cosine similarity, and then determined whether there is a metaphor or not by the similarity between the matching word and the target word. We use the pre-trained version of CBOW (Mikolov et al., 2013) in 100 dimensions on Wikipedia and Gigaword corpus[1]. If the similarity between either target word and the subject or object is greater than 0, it is determined to be a metaphor. Shutova et al. (2016) also used cosine similarity, but only considered the similarity between the verb and the subject or object. We use the same pre-trained model of CBOW. Again, similarity greater than 0 is judged as metaphorical. Turney et al. (2011) adopted abstraction degree for metaphorical judgment, which assumes that relatively abstract words paired with relatively concrete words produce metaphors. We use the abstraction degree ratings (Brysbaert et al., 2014) to determine SO and VO pairs with relatively abstract relationships as metaphors (a rating difference greater than 0.5 is recognized as relatively abstract relationship). To ensure a fair comparison, we use the subject-predicate-object extracted by ChatGPT as the prepositioned subject and object of the target word in context.

Experiment 2. Experiment 2 compares our unsupervised method with the zero-shot or few-shot sample generation methods designed by us. The sample generation methods first uses ChatGPT to generate metaphor samples (See Appendix §12 for the specific prompts used), and then fine-tuned using a pre-trained model. Specifically, we employ two different prompts: one is Direct Generation (DG) and the other is Example Prompt Enhancement (EPE). EPE provides a manually labeled example for each sample given the verb and label (metaphorical or literal). Labeled data from

[1]https://huggingface.co/fse/glove-wiki-gigaword-100

Models	TroFi				MOX-H			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
WordNet_Topic	46.0	96.8	44.6	**61.0**	53.6	90.1	51.4	65.4
WordNet_Topic_k	46.2	95.9	44.5	60.6	54.1	88.6	51.7	65.3
LDA_Topic	45.9	91.4	44.2	59.6	51.2	94.0	50.0	65.3
LDA_Topic_k	44.5	96.9	43.9	60.4	52.2	92.9	50.3	65.3
Oxford_Topic	47.0	90.4	44.6	59.8	62.9	86.7	58.1	69.6
Oxford_Topic_k	45.8	93.7	44.2	60.1	61.2	93.3	56.1	**70.1**

Table 4: Performance comparison on MOH-X and TroFi datasets using different topic mappings. The WordNet_Topic, LDA_Topic, and Oxford_Topic represent three different topics, respectively. The ones ending with "k" indicate that the first 3 nouns are extracted as the center nouns, while the ones without "k" indicate that first 1 noun is extracted.

the VUAverb training set was randomly selected as examples for EPE. The samples generated by both DG and EPE were fine-tuned using RoBERTa-large.

5 Results and Discussion

Experiment 1. From the results in Table 2, all our methods achieve the best performance. On the three datasets, our methods improves 29.5%, 40.8% and 37.3% on the core metric F1 compared to GPT-3.5 Turbo, respectively. This suggests that the surface knowledge generated by bootstrapping and combining GPT can significantly improve GPT's performance in detecting verb metaphors. In addition, compared with unsupervised strong baseline (SIM-CBOW), our method improves the performance on the three datasets by 3.3%, 0.8% and 5.9%, respectively. This demonstrates the superiority of our unsupervised approach. However, compared to the TroFi and MOH-X datasets, all methods perform poorly on VUAverb. The possible reason for this is that VUAverb (989 verbs) contains a larger and wider range of verb types compared to TroFi (68 verbs) and MOH-X (215 verbs), which requires unsupervised methods to explore more knowledge. For example, in our approach, the verb list needs to expand the verb types to 989, and each verb needs to guide ChatGPT to generate the corresponding literal collocation. The above approach introduces noise while increasing the coverage of the verb list.

Experiment 2. The results of comparing with the sample generation methods are shown in Table 3. There is still a gap between the performance of EPE and our unsupervised method on MOH-X and TroFi. Our unsupervised method obtains a 12.7% performance improvement on MOH-X, which further proves the superiority of our method. In addition, our unsupervised method is slightly lower than EPE (3%) since the labeling examples used in the EPE method are derived from VUAverb.

6 Topic Experiment

We examined the impact of the three topic mappings introduced in §3.2 on model performance. For WordNet topics, we use the NLTK library in Python to extract the superlatives and synonyms of the central noun, and then combine all of them into the WordNet topics set corresponding to the target verb. For LDA topics, we use WUPS (Shet et al., 2012) to calculate the similarity between the central noun and the 60 LDA topics words, and classify them into one or more LDA topics based on the similarity. For Oxford topics, we first access the Oxford lexicon for pronoun disambiguation and named entity conversion, and then convert them into one or more topic categories corresponding to the Oxford lexicon.

Specifically, we first parse the input text to extract the subject and object corresponding to the target verb. We select by default the first k nouns as the subject content to be converted (k is a hyperparameter). We consider the case of extracting 1 or 3 central nouns. Specific topic types include WordNet_Topic, WordNet_Topic_k, LDA_Topic, LDA_Topic_k, Oxford_Topic, Oxford_Topic_k, where k means extracting the first k nouns as the center nouns.

As shown in Table 4, the three topic types performed relatively close to each other on the TroFi dataset, with the WordNet topics achieving the best performance with an F1 score of 61.0%. On the

MOX dataset, the WordNet topics and the LDA topics perform similarly, while the best performance is obtained using the Oxford Dictionary topic, with an F1 score of 70.1%, which is 4.8% higher than the other two topics. Regarding the hyperparameter k, we observed that setting k to 1 or 3 did not have a significant performance difference between the two datasets when using either the WordNet topics or the LDA topics. However, setting k to 3 slightly improves the performance when using the Oxford topics. This may be due to the fact that there is polysemy in Oxford topics, i.e., different noun meanings correspond to multiple topic information, which extends the scope of the verb list to cover literal topics.

7 Hyper-parameter Experiment

To balance the set size with the metaphors detection accuracy when introducing topic sets, we introduce two additional hyperparameters for control. Specifically, k_1 represents the number of literal or non-metaphorical collocations selected from the verb list, while k_2 denotes the number of topics that may be covered by the subject and object corresponding to the target verb. Larger values of k_1 imply that the model's predictions cover more literal-meaning collocations of verbs, while larger values of k_2 indicate that more meanings of the subject- or object-centered words are used in the metaphorical relations parsed in the text.

In this regard, the hyper-parameter experiment aims to explore the effect of two hyper-parameters, k_1 and k_2, on the model metaphor detection performance. Considering the results of the previous topic experiment, we find that Oxford_Topic_k, which extracts 3 central nouns, performs better relative to Oxford_Topic_k, which extracts 1 central word. Moreover, when only 1 central noun is extracted, there are relatively fewer topic types (which depends on the number of different meanings of that central noun). Specifically, the hyper-parameter experiment will fix the hyper-parameter of the center word as $k = 3$, while setting the value range of k_1 and k_2 between 1 and 9. In addition, the experiments will be conducted on the MOH-X.

Detailed results can be found in Figure 1. On the one hand, the model performance improves as the value of the hyperparameter k_1 increases. This can be attributed to the fact that increasing k_1 introduces more literal collocations from the verb list. As a result, the model is more capable of detecting the non-metaphorical content associated with a particular verb and reduces misclassification. On the other hand, the performance peaks when the hyperparameter k_2 is set to 3. However, when continuing to increase the value of k_2, the model's performance in detecting metaphors decreases instead. This suggests that considering multiple meanings of the central word may introduce metaphorical information or redundant topics. Thus, our experimental results emphasize the need to weigh the model performance and the impact of topic introduction when choosing the value of k_2.

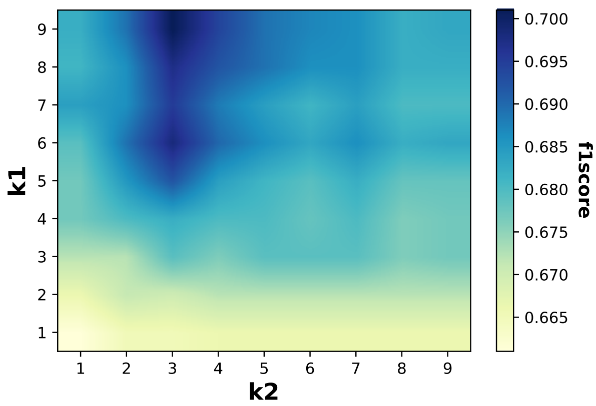

Figure 1: Effect of parameters $k1$, $k2$ on model performance, where k_1 represents the number of literal or non-metaphorical collocations selected from the verb list and k_2 denotes the number of topics that may be covered by the subject and object corresponding to the target verb.

8 Conclusion

We present a novel approach aimed at improving the performance of unsupervised verb metaphors detection task using ChatGPT. This approach does not rely on hand-coded knowledge or manually labeled datasets. First, we construct a literal meaning collocation lookup list for each target verb. When parsing the input text, we pay special attention to the subjects and objects corresponding to the verbs to be detected. We introduced a variety of topics, including WordNet topics, LDA topics, and Oxford topics. By comparing the relationship between subject and object topics in the input text and the verb topics in the verb list, we determine whether the text contains metaphorical expressions. The results show that by delicately combining and directing the world knowledge, we are able to significantly improve the performance of ChatGPT in the verb metaphors detection task.

9 Limitations

We introduce a verb list containing literal subject-verb and verb-object collocations for each target vocabulary. However, the literal collocations generated using ChatGPT are not always comprehensive, which leads to some literal samples being incorrectly categorized as metaphorical usage. In addition, due to varying syntactic structures, when analyzing subject-verb-object relations in input texts using ChatGPT, there may be parsing errors or structures that are not present, which also affects the performance of the overall method. In future work, we would like to investigate more powerful generative models or natural language parsing tools to improve the coverage of literal collocations in verb lists or to improve the accuracy of parsing subject-verb-object relations of input texts.

10 Ethics Statement

Metaphor, as a linguistic phenomenon that conveys implicit semantics, is capable of concretizing abstract concepts or enriching substantive concepts. This makes it possible for metaphors to be used as a tool for communicating political positions and gaining voter support in the political domain. However, our proposed zero-shot metaphors detection approach can also be used to identify metaphorical expressions and address the above issues from a governance perspective. In addition, we advocate the inclusion of tasks related to metaphors detection and generation, especially the application of ChatGPT to downstream metaphor applications, into the AI ethical code.

References

Mateusz Babieno, Masashi Takeshita, Dusan Radisavljevic, Rafal Rzepka, and Kenji Araki. 2022. Miss roberta wilde: Metaphor identification using masked language model with wiktionary lexical definitions. *Applied Sciences*, 12(4):2081.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46:904–911.

Erik Cambria, Soujanya Poria, Alexander Gelbukh, and Mike Thelwall. 2017. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6):74–80.

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. Bllip 1987-89 wsj corpus release 1. *Linguistic Data Consortium, Philadelphia*, 36.

B Edition, BNC Baby, and BNC Sampler. British national corpus.

Lisa Gandy, Nadji Allan, Mark Atallah, Ophir Frieder, Newton Howard, Sergey Kanareykin, Moshe Koppel, Mark Last, Yair Neuman, and Shlomo Argamon. 2013. Automatic identification of conceptual metaphors with limited knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 328–334.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Ilana Heintz, Ryan Gabbard, Mahesh Srivastava, Dave Barner, Donald Black, Majorie Friedman, and Ralph Weischedel. 2013. Automatic extraction of linguistic metaphors with lda topic modeling. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 58–66.

Mary Hesse. 1965. Models and analogies in science.

Yael Karov and Shimon Edelman. 1998. Similarity-based word sense disambiguation. *Computational linguistics*, 24(1):41–59.

Adam Kilgarriff. 2000. Wordnet: An electronic lexical database.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430.

Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting elusive metaphors using lexical resources. In *Proceedings of the Workshop on Computational approaches to Figurative Language*, pages 13–20.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 vua and toefl metaphor detection shared task. In *Proceedings of the second workshop on figurative language processing*, pages 18–29.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 vua metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66.

Rui Mao, Chenghua Lin, and Frank Guerin. 2018. Word embedding and wordnet based metaphor identification and interpretation. In *Proceedings of the 56th annual meeting of the association for computational linguistics*. Association for Computational Linguistics (ACL).

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33.

Malay Pramanick and Pabitra Mitra. 2018. Unsupervised detection of metaphorical adjective-noun pairs. In *Proceedings of the Workshop on Figurative Language Processing*, pages 76–80.

KC Shet, U Dinesh Acharya, et al. 2012. A new similarity measure for taxonomy based on edge counting. *arXiv preprint arXiv:1211.4709*.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 160–170.

Ekaterina Shutova and Lin Sun. 2013. Unsupervised metaphor identification using hierarchical graph factorization clustering. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 978–988.

Ekaterina Shutova and Simone Teufel. 2010. Metaphor corpus annotated for source-target domain mappings. In *LREC*, volume 2, pages 2–2. Citeseer.

Ekaterina Shutova, Tim Van de Cruys, and Anna Korhonen. 2012. Unsupervised metaphor paraphrasing using a vector space model. In *Proceedings of COLING 2012: Posters*, pages 1121–1130.

Wei Song, Shuhui Zhou, Ruiji Fu, Ting Liu, and Lizhen Liu. 2021. 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)*, pages 4240–4251.

Gerard Steen, Aletta G Dorst, J Berenike Herrmann, Anna Kaal, Tina Krennmayr, Trijntje Pasma, et al. 2010. A method for linguistic metaphor identification. *Amsterdam: Benjamins*.

Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690.

Lennart Wachowiak and Dagmar Gromann. 2023. Does gpt-3 grasp metaphors? identifying metaphor mappings with generative language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1018–1032.

Shun Wang, Yucheng Li, Chenghua Lin, Loïc Barrault, and Frank Guerin. 2023. Metaphor detection with effective context denoising. *arXiv preprint arXiv:2302.05611*.

Yorick Wilks, Adam Dalton, James Allen, and Lucian Galescu. 2013. Automatic metaphor detection using large-scale lexical resources and conventional metaphor extraction. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 36–44.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Shenglong Zhang and Ying Liu. 2023. Adversarial multi-task learning for end-to-end metaphor detection. *arXiv preprint arXiv:2305.16638*.

11 Appendix A

The main purpose of this section is to detail how LLaMA2 or GPT3.5-Turbo can be utilized to obtain literal collocations of verbs, as well as to obtain the required prompt for subject and object pairs in the input text.

11.1 Analyzing Literal Collocations

For verb literal collocation parsing, we assume that the target verb is w_k. We do this by making a request to LLaMA2 or GPT3.5-Turbo to generate all possible literal collocations of w_k, including both subject-predicate and predicate-object parts. We explicitly labeled the desired output format at the end of the request:

> Please provide as many subject and object topic categories as possible that are paired with the verb '$\omega_\mathbf{k}$' in non metaphorical or literal usage. The format is: Subject Categories:
> 1.
> 2.
> Object Categories:
> 1.
> 2.

11.2 Analyze Subject-Object Pairs

For subject-object parsing of the input text, we consider a specific target verb w_k, whose corresponding context is S, and the position of the verb w_k in the context is indicated by the index k. We make a request to GPT3.5-Turbo to generate the subject and object corresponding to the verb w_k in the context. Again, we explicitly labeled the desired output format at the end of the request:

> For the sentence '$\mathbf{S}$'. Give the subject and object of the verb '$\omega_\mathbf{k}$' located in '$\mathbf{k}$' in order of format. For example,
> subject:
> object:

12 Appendix B

This section presents the prompts used in the ChatGPT-based Direct Generation (DG) and Example Prompt Enhancement (EPE) methods in Experiment 2. **n** represents the number of samples to be generated, and this number is related to the distribution of the VUAverb training set. w_k represents the target word. Based on the specified label (metaphorical or literal), ChatGPT is guided to generate the context of the word to reflect its metaphorical or non-metaphorical usage. In EPE, additional examples (randomly selected from the VUAverb training set) are required for each target word w_k and specified label.

> **DG:**
> Generate **n metaphorical** sentences of different styles based on the given verb. Each sentence must contain the given verb and be output after s-1 to s-**n** respectively.
> verb: $\omega_\mathbf{k}$
> s-1:
> ......
>
> **EPE:**
> Generate **n metaphorical** sentences of different styles based on the given verb, imitating the example. Each generated sentence is to contain the given verb and is to be output after s-1 to s-**n** respectively.
> verb: $\omega_\mathbf{k}$
> example: **example**
> s-1:
> ......