A study in metaphor generation: fine-tuning BERT to classify adjectives as literal or metaphorical

A paper for Language Engineering Applications [HOT29a]

Austin Moore

Table of Contents

1. Introduction **3**

2. Complexity of metaphor **4**

3. Three metaphor generation models **7**

3.1 Yu & Wan (2019) 7

3.2 Stowe, Ribeiro, Gurevych (2020) 8

3.3 Chakrabarty, Zheng, Muresan, Peng (2021) 9

4. An adjectival metaphor model **12**

5. Conclusion **14**

Bibliography **15**

1. Introduction

The concept of teaching a computer to understand metaphor sounds at first like something that should be confined to science fiction or the far-flung future. This is because metaphor in language is a difficult concept to grasp and often eludes clear categorisation. It is also ubiquitous such that clear delineation between literal and metaphorical language, on which the task of metaphor generation relies, is a complex task that needs to be approached scientifically – in a precise and non-metaphorical way. The issue is that if we do hope to reach a point where human-machine linguistic interaction is seamless and quasi-natural, we cannot ignore the need to infuse natural language processing with metaphorical sensibility.

Although this is an area that has seen comparatively little research in relation to other domains of NLP, there have been nevertheless some impressive developments, achieved by approaching the problem as incrementally as it needs to be. These range from unsupervised metaphor identification to systems that take a sentence with a literal verb as input and output a metaphorical replacement for the verb in that sentence.

In this paper, I will first elaborate on the problem of rigid metaphor classification and the approaches some have taken to solve it. I will then draw attention to the efforts that have been made to bring the task of intelligent metaphor generation to fruition. Finally, I will describe my own small contribution to this field in the form of an adjectival metaphor corpus inspired by Chakrabarty et al. (2021) and using the VU Amsterdam Metaphor Corpus (Steen at al, 2010).

2. Complexity of metaphor

I will here describe the difficulty of metaphor identification and outline two methods that have been developed to address this. Current efforts in metaphor generation rely variously on annotated datasets so it is worth drawing some attention to how a prominent example is constructed. However, as will be described in the next chapter, there have recently been methods developed which can identify and interpret metaphor using unsupervised neural networks.

The first step towards metaphor generation is metaphor identification. The potential complexity of this can be illustrated by the many forms in which linguistic metaphor can be applied. In the introduction to this paper I used the phrase *far-flung future* whose clear adjectival metaphor *far-flung* invokes distance and exoticism through the suggestion of a physical action. In the following sentence I wrote *metaphor in language is a difficult concept to grasp*. The metaphor of grasp is used in many languages in order to convey a sense of understanding (consider the list of Romance derivatives of Latin *comprehendere* and cognates of *be-greifen* in Germanic languages (McGilchrist, 2009, p136)). However this is less clear than far-flung because it is an example of a metaphor described by Pawelec (2007) as 'dead' – i.e. as having in this context a conventional meaning different from its original meaning. These are just two examples I have pointed out to convey the difficulty of classification and the conceptual layers involved in every metaphor. The developers of the Metaphor Identification Procedure (MIP) (Pragglejaz Group, 2007) – the first formalised and systematic procedure for metaphor identification in language usage – tried to avoid this complexity by proposing a set of rules to determine simply whether a lexical unit was metaphorical or not:

1. Read the entire text/discourse to establish a general understanding of the meaning.
2. Determine the lexical units in the text/discourse.
3. a. For each lexical unit in the text, establish its meaning in context, i.e. how it applies to an entity, relation or attribute in the situation evoked by the text (contextual meaning). Take into account what comes before and after the lexical unit.

b. For each lexical unit, determine if it has a more basic contemporary meaning in other contexts than the one in the given context. For our purposes, basic meanings tend to be:

– more concrete; what they evoke is easier to imagine, see, hear, feel, smell, and taste;
– related to bodily action;
– more precise (as opposed to vague);
– historically older.

Basic meanings are not necessarily the most frequent meanings of the lexical unit.

c. If the lexical unit has a more basic current/contemporary meaning in other contexts than the given context, decide whether the contextual meaning contrasts with the basic meaning but can be understood in comparison with it.
4. If yes, mark the lexical unit as metaphorical.

While this framework was designed as a straightforward method for metaphor identification, as Steen et al. (2010) point out, it entailed two significant weaknesses. The

first is that there are many decisions that still need to be made by analysts in order to identify words related to metaphor. This means that if two analysts' opinions differ, they will disagree on whether the use of a lexical unit – itself an ill-defined concept – counts as metaphorical. This is clearly not ideal, as the goal of a classification system should be rigidity, leaving little to no room for disagreement. The second weakness of MIP is that it only identifies the linguistic form of metaphor and not the underlying conceptual structures. These complexities are important for the purposes of metaphor understanding and generation because the more shades of information can be supplied about a metaphor use – not simply whether it is or is not metaphorical – the more informed a system will be in semantic mapping between literal and metaphorical.

To address these issues, MIPVU (Steen et al., 2010) was developed, whose authors claimed that by following the following set of guidelines, metaphor identification could be achieved in a systematic and exhaustive way:

1. Find metaphor-related words (MRWs) by examining the text on a word-by-word basis.
2. When a word is used indirectly and that use may potentially be explained by some form of cross-domain mapping from a more basic meaning of that word, mark the word as metaphorically used (MRW).
3. When a word is used directly and its use may potentially be explained by some form of cross-domain mapping to a more basic referent or topic in the text, mark the word as direct metaphor (MRW, direct).
4. When words are used for the purpose of lexico-grammatical substitution, such as third person personal pronouns, or when ellipsis occurs where words may be seen as missing, as in some forms of co-ordination, and when a direct or indirect meaning is conveyed by those substitutions or ellipses that may potentially be explained by some form of cross-domain mapping from a more basic meaning, referent, or topic, insert a code for implicit metaphor (MRW, implicit).
5. When a word functions as a signal that a cross-domain mapping may be at play, mark it as a metaphor flag (MFlag).
6. When a word is a new-formation coined, examine the distinct words that are its independent parts according to steps 2 through 5.

A detailed explanation of how metaphors are annotated by MIPVU can be found here. This method has formed the basis for the VU Amsterdam Metaphor Corpus Online (VUAMC), which two of the three of the major models in metaphor generation have used. It is the largest available corpus for complex metaphorical language use and includes 190,000 lexical units from a subset of four broad registers from BNC-baby. Such large datasets take a long time to compile, and they are still nowhere near of adequate volume for training large neural network architectures. These issues among others have motivated approaches like that of Mao et al. (2018) who have attempted to employ word embedding-based models that identify metaphorical words using their surrounding context within a sentence. Such methods have produced impressive results, but cannot yet come close to being able to encode the same informational depth for each metaphor as a comprehensive annotation system like the VUA. Similar to the problem that motived the introduction of MIPVU over MIP, such systems do not capture the conceptual scaffolding that underlie how the metaphors they identify are used. I will elaborate on this the next chapter.

While metaphor identification systems represent the first step towards metaphor generation, there are more challenges to address. For example, in order to train a model to suggest literal substitutions for metaphorical language use or vice versa, parallel corpora will be required. Such corpora are rare and the ones that do exist are not large enough to train a model like BERT (e.g. see Mohammed et al., 2016). This means that researchers have needed to approach this requirement almost from scratch for each the three models I will now describe.

3. Three metaphor generation models

3.1 Yu & Wan (2019)

The first major attempt at training an end-to-end model to generate metaphor was by Yu & Wang (2019), whose goal was to generate sentences containing metaphorical verbs in particular. The authors generated their own metaphor corpus by taking phrases from the English Wikipedia corpus whose cosine similarity between words are lower than a certain threshold. This is based on the approach of Shutova et al. (2016) and Rei et al. (2017) whose work attempted to bypass the need for manual metaphor identification by showing that the metaphoricity of particular phrases could be derived in the dissimilarity of their embeddings. The intuition here is that when both words in a particular two-word phrase are literal, they will come from the same domain and therefore their embeddings will be closer than when one of them comes from a different source. An example of this is in the phrase *colourful person*, where the adjective *colourful* is taken from another domain to metaphorically describe *person*. The limitations of this approach is that it can only identify metaphor at phrase-level. This means that when given the a sentence like '*She devoured his novels'*, the system will parse the sentence into a verb-object phrase *devour novel* and will then identify a disparity between *devour* and *novel.* It will flag the phrase as a likely metaphor, but will not be able to provide any more detailed information as to the nature of the metaphor. It also discards other context within the sentence (Mao et al., 2018). A final weakness of this approach is that it will fail to detect when a two-word phrase is metaphorical when both words come happen to come from the same domain, as in *climb ladder*.

Once a metaphorical verb-phrase is identified, Yu & Wang's model generates a list of candidate literal-sense synonyms from which it chooses that of the highest suitability in the given context. It then pairs the metaphorical verb with a literal one to create parallel training data.

Yu & Wang's approach was inspired by seq2BF, (Mou et al, 2016) which generates a reply containing a given keyword. On a high very high level, this works by combining a backward and forward seq2seq models which are based on recurrent neural networks using gated recurring networks (GRUs). Using this, they implemented a part of speech (POS) constrained language model coupled with an adjustable joint beam search algorithm (Post & Vilar, 2018) which outputs sentences containing an assigned verb, in this case a metaphorical one. Essentially, what is happening here is that when the model is given a target metaphorical word, it will use what it has learned about literal pairings to assign a fit word. It will then generate a sentence in which the metaphorical verb will carry the same contextual meaning as the fit word. An example of a successful output highlighted in the paper is the following. When provided with the word *absorbed* as input, the model decided on *learned* as the fit word and output the sentence *he absorbed his studies at the University of Birmingham* . This is quite an interesting result, but it is clearly borne from a limited approach to metaphor generation which relies on highly constrained parameters: two-word phrases, direct metaphors (in the MIPVU sense) and phrase-level metaphor parsing only.

3.2 Stowe, Ribeiro, Gurevych (2020)

I will now turn to a another approach proposed by Stowe et al. (2020) who set out to achieve metaphoric paraphrase generation. This model took a similar approach to Yu & Wang but focused on the more constrained task of outputting a novel verb-oriented metaphorical paraphrase when given a literal phrase as input. They proposed two models.

The first model is a lexical replacement approach which again relies on an unsupervised approach to identify metaphor from its context within a sentence. This time however, it is based on the work of Mao et al. (2018) who improved upon the phrase-level approach described above by now having the capacity to identify metaphor at word level. Stowe et al. took this method designed to identify metaphor and reversed it by using it instead to identify literal verbs with the goal of replacing them with metaphorical ones. They then used the WordNet (Princeton University, 2010) sense hierarchy to identify troponyms as replacement candidates. This was a clever touch, as they rightly point out that replacement with more specific verbs is 'likely to yield more metaphoric expression, as these specific verbs require specific contexts to be understood literally' (Stowe et al., 2020, p4). Finally, the best fitting candidate is chosen to generate metaphor. An example of success for this lexical replacement approach is '*The moon **sparkled** back at itself from the Lake's surface*' being generated from the input '*The moon **reflected** back at itself from the Lake's surface*'.

The second model proposed by Stowe at al uses a seq2seq masked language model (MLM). In order to create artificial parallel training data, a 'metaphor masking' approach is used which replaces metaphoric words with a unique 'metaphor' tokens. Through training, the model learns it needs to generate a metaphorical word when it encounters these metaphor tokens. In testing – material for which came from the same dataset (i.e., Mohammed et al., 2016) as the training – the literal input is given, while the verb is masked. This steers the model to replacing the literal verb with a metaphorical paraphrase.

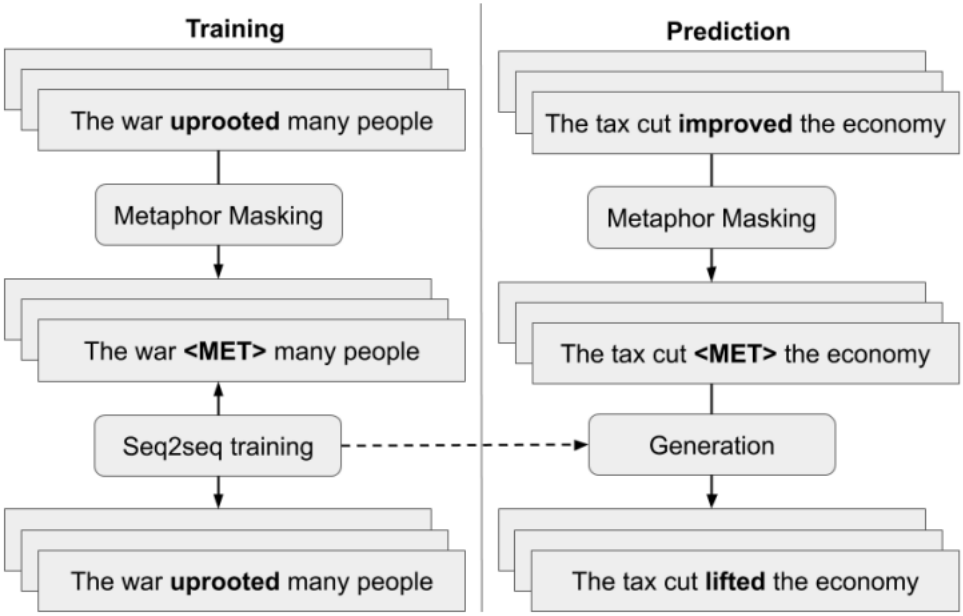

Figure 1: Metaphor masking method (Stowe et al., 2020)

This elegant approach does however require annotated data to compile its parallel training data. Drawing from VUAMC plus three other smaller datasets, the total number of verbs is 35,415 of which 11,593 are metaphoric. In spite of some good results coming from this model, this number is not large enough to effectively train a neural network. Another issue with this masking method is that it does not learn the semantic mapping between literal and metaphorical verbs (Chakrabarty et al., 2021). In figure 1 above for example, when the model encounters '*The tax cut* <***MET***> *economy',* it 'understands' that it needs to replace the metaphor token with a metaphorical verb that fits the context of the sentence. However, the choice of *lifting* it is not due to any relationship this verb in particular has with the original literal verb *improved*. These two issues – lack of training data and lack of semantic mapping between literal and metaphorical verbs – will be addressed by the next and perhaps most successful metaphor generation model to date.

3.3 Chakrabarty, Zhang, Muresan, Peng (2021)

Chakrabarty et al. propose MERMAID: Metaphor Generation with Symbolism and Discriminative Coding. The goal here is to expose a model to an unseen sentence and have it replace a literal verb with a metaphorical one. The authors devised a method to automatically construct a parallel corpus containing 93,498 literal-metaphorical sentence pairs. They do this by fine-tuning BERT (Devlin et al., 2018) on VUAMC to classify verb metaphors. Fine-tuning BERT does not require nearly as much data as training it from scratch (Hao et al., 2019; Sun et al., 2020), so VUAMC serves well here particularly when training for verbs.

They apply the classifier to the Gutenberg Poetry Corpus which contains over 3 million lines of poetry. They then employ BERT's ability to predict a masked word by learning the context of the surrounding words. They implement this similarly to Stowe et al. who

used the same technique to produce literal counterparts to metaphorical verbs. However, here, Chakrabarty seek to address two weaknesses in using masked language model (MLM) to generate literal verbs: '1) the output will not necessarily be literal; 2) after replacing the default MLM-predicted verb, the metaphorical sentence and the new sentence with the replaced verb might be semantically dissimilar.'

To address the first issue, they simply rerank predicted verbs based on literal scores so that the verbs deemed most literal will be chosen first. Addressing the second issue is more complicated, as a method needed to be devised to force the model to choose a literal replacement verb that bestows the sentence the same semantic sense as the metaphorical one did before it. An example of when this does not happen is as follows. Consider the case where the model sees the sentence '*The turbulent feelings that surged through his soul*' and replaces the word *surged* with *eased* to output the following: '*The turbulent feelings that eased through his soul'*. Here we have two sentences which due to their verbs have quite different senses. Surging suggests a quick and heavy sensation whereas easing suggests the opposite. In order to ensure the semantic mapping between literal and metaphorical verb is preserved, the model employs COMET (Bosselut et al., 2019). This is a transformer language model whose designers claim is capable of learning 'commonsense knowledge' about language. When humans interpret language, they are able to bring broader knowledge to texts that allow them to be able to understand complex metaphorical relationships. It makes sense to want that deep learning-based metaphor generation models be taught to learn such broader knowledge as much as is feasible. MERMAID uses COMET's ability to associate a word with a symbol – e.g. *dove* being associated with *purity* – to infuse each lexical unit in the training set sentences with a corresponding symbol. It can then ensure the suggested literal verb retains the same sense as the metaphorical one. To ensure the highest quality of output, the authors force the model to choose a replacement verb that brings the same top-5 suggested tokens (COMET decides this with a beam search of width 5) as the metaphorical verb did.

Meta Input	The turbulent feelings that **surged** through his soul .	
Inp Symbol	love, loss, despair, sorrow, loneliness	
Lit Output1	The turbulent feelings that *eased* through his soul .	✗
Symbol	peace,love,happiness,joy,hope	
Lit Output2	The turbulent feelings that *continued* through his soul .	✓
Symbol	love, loss, despair, sorrow, loneliness	

Table 1: An example of an input sentence along with the input symbols that COMET has associated with the verb '*surged*'. Lit Output1 is a rejected candidate since all 5 symbols associated with '*eased*' do not match those of the input symbols. (Charabarty et al., 2021)

Following this strict filtering process, 90,000 pairs are used for training and an additional 3,498 for validation.

Now that the training data has been compiled, the authors face the same two problems for generating metaphors in unseen sentences. Again, these are 1) ensuring that the replacement verb is actually in the desired category – in this case metaphorical – and 2) ensuring the metaphorical replacement verb carries the same sense as the literal one. They tackle the second issue by fine-tuning BART (Lewis et al., 2019) on the training data, treating the literal input as encoder source and the metaphorical output as decoder target. To force metaphorical output – rather than literal which various pre-trained language models tend to be biased towards – they then use a discriminative decoding system trained by fine-tuning RoBERTa (Liu et al., 2019) on VUAMC and on another annotated dataset by Beigman Klebanov et al. (2018).

Combining BART and RoBERTa, the discriminative decoding process is the following. A sentence with a literal-sense verb is inputted. BART outputs the top-5 metaphor-sense verb candidates having the same sense as the literal one. RoBERTa then classifies the sentence as literal or metaphorical based on whether or not a truly metaphorical verb is present. The 5 BART suggestions are then reranked according to which one is most likely to contain an actual metaphorical verb.

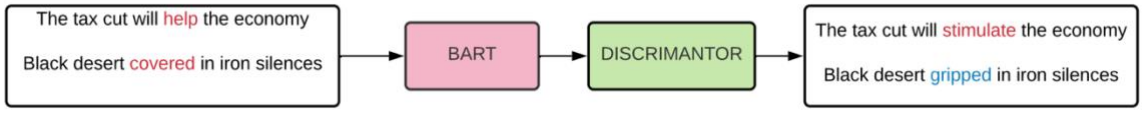

Figure 2: This depicts the process of first using BART to have the outputted metaphor verb candidate match the semantic sense of the literal input, then using the roBERTa-based discriminator to choose the candidate with the highest metaphoricity score. (Chakrabarty et al., 2021)

A further element that makes MERMAID more ambitious than Stowe et al. (2020) is that its test data was derived 'from the wild' – from sources totally divorced from its train set. The authors even went so far as to ensure diversity in genre by scraping the subreddits WritingPrompt – which would feature prose language – and OCPoetry – obviously poetry. They included sentences of up to 12 words in length which were deemed as containing literal verbs by the previously described fine-tuned BERT model which had been used to create the parallel training data. MERMAID then outputted significantly higher quality metaphors – judged both automatically and by human experts based on various criteria – than Stowe et al. whose metaphor masking model is also tested for direct comparison.

And the hills have a shimmer of light between, And the valleys are *covered* with misty veils, And ....., ...
And the hills have a shimmer of light between, And the valleys are *wrapped* with misty veils, And ....., ...
Leaves on a maple, *burst* red with the shorter days; Falling to the ground. ....
Leaves on a maple, *burgeoned* red with the shorter days; Falling to the ground. ....

Table 2: Samples from r/OCPoetry for which MERMAID generated replacement metaphorical verbs (in blue). (Charabarty et al., 2021)

While the results here are impressive, it is worth noting how highly complex MERMAID is to put together. It weaves together four different language models – fine-tuned BERT, COMET, fine-tuned BART and fine-tuned roBERTa – in ingenious yet convoluted ways only to be able to replace literal verbs with metaphorical ones. This is another testament to how difficult a task metaphor generation is, and how much progress is still to be made.

4. An adjectival metaphor model

All three of the models I discussed above focused on verb metaphors. The reasons the authors of these papers made this choice is that they are confident that the most common types of metaphors are expressed in verbs. While this may be the case, if we want to design intelligent and more comprehensive metaphor generation systems, it is of course important to study other parts of speech. We have tried to take influence from MERMAID and fine-tune BERT to predict on adjective metaphors. If successful, this could then be used as a classifier to create a parallel training corpus using for example the Gutenberg Poetry corpus as Chakrabarty et al. did with verb metaphors.

Mao et al. (2019) developed train, validation and test sets of VUAMC. We parsed these by placing [CLS] tokens – used for fine-tuning BERT for sequence classification – before each sentence, allowing us to analyze the dataset at sentence level. Each sentence carries useful information such as the POS tags of each lexical unit – meaning we can detect whether an adjective is present as well its position in the sentence – as well as whether each word is metaphorical. Thus we first detect whether a sentence contains at least one adjective. The next step is that we extract two extra pieces of information for each adjective-containing sentence: the index position of the adjective, and whether or not it is metaphorical (0 for literal; 1 for metaphorical). If there are more than one adjective per sentence, we will parse that sentence as many times as there are adjectives, each time focusing on a different adjective. Take for example the sentence '*Now, however, the wheel has turned full circle'*. Here the adjective '*full*' would be identified at index position 12 (taking into account the [CLS] token at the beginning as well as the punctuation marks) and be identified as metaphorical with a 1. We repeat this process for each sentence in the train, validation and test sets. We end up with 9673 samples in the train set – 1115 with metaphorical adjectives and 8558 with literal ones. In the validation set we have 3213 samples of which 344 are metaphorical. In the test set we have 3965 samples of which 541 are metaphorical.

We then tokenize our datasets, ensuring that all of our individual datapoints are padded to the same length and truncated which will allow us to evenly feed each one to the model. The next step is to take the tokenised data and convert it to token type id vectors. These are all-zero vectors except for the position of the adjective, which takes a 1. This means the model will learn which position in a sequence it needs to focus on in order to make a prediction. Also encoded in this data structure is the label of the adjective in each sequence: metaphorical or not metaphorical.

Next we fine-tune a BERT model using our train and validation sets. Predicting on the train set, we see encouraging results. The overall accuracy is around 91%. This does not tell us much however, as the model will mostly decide that an adjective is used literally and will indeed be right most of the time given the prevalence of literal uses in the dataset (3424 literal vs 541 metaphorical). Therefore the precision and recall of the minority class – metaphorical usages – will give us the answers we are really interested in. The precision comes to about 79%, meaning that when a metaphorical usage is predicted, the model is correct around 3 out of 4 times. The recall meanwhile comes to about 50%, meaning that the model only recognises about half of the metaphorical adjectives in the

test set. Taking the harmonic mean of the precision and recall results in an F1 score of about 0.6.

Accuracy	Precision	Recall	F1
91%	79%	50%	0.61

Table 3: Output metrics of our adjective classifier model upon its application to the test set

We then compared this to results when parsing for verbs instead of adjectives. The first thing to note is that there are far more verbs in VUAMC than adjectives, leading to a total of 20917 samples in the train set, 7152 in the validation set and 9872 in the test set. We would expect that this considerably larger dataset would lead to better results and indeed while the accuracy is the same at around 91%, the precision and recall are improved at 77% and 67% respectively for an F1 score of 0.7.

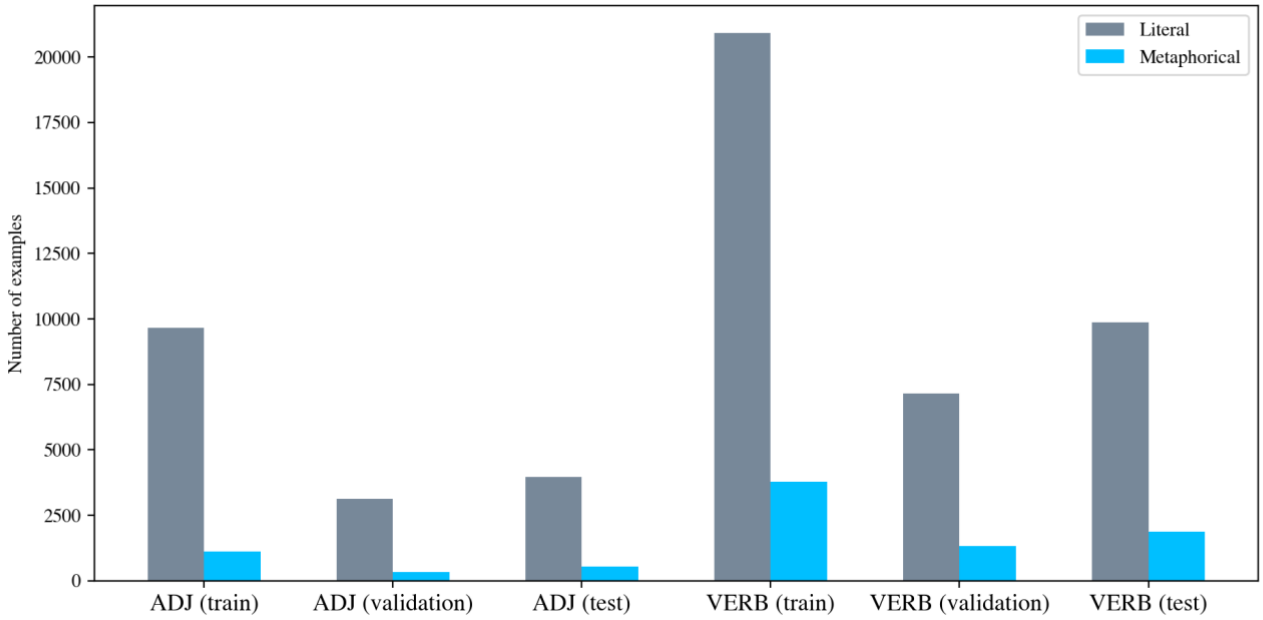

Figure 3: Comparison of the quantity and composition of train, validation and test sets (developed by Mao at al. (2019)) derived from VUAMC parsed for adjectives in one case and verbs in the other.

Another reason the fine-tuned BERT model performs better trained on verbs than on adjectives may be that the verb dataset has a higher proportion of metaphorical verbs to literal verbs than does the adjective dataset.

	Adjectives (16761)			**Verbs** (37941)		
	Train	Validation	Test	Train	Validation	Test
Total	9673	3123	3965	20917	7152	9872
Literal	8558 **(88%)**	2869 **(92%)**	3424 **(86%)**	17132 **(82%)**	5812 **(81%)**	7985 **(81%)**
Metaphorical	1115 **(12%)**	344 **(8%)**	541 **(14%)**	3785 **(18%)**	1340 **(19%)**	1887 **(19%)**

Table 4: Demonstration that the ratio of metaphorical to literal examples is higher in the verb datasets than in those of the adjectives

Overall, what we have done is to investigate the results when fine-tuning BERT to classify metaphors in a part of speech in which they appear to be less common than for verbs, where the research in metaphor generation has focused up to now. The results are good but not as good as when training for verbs essentially due to the fewer examples of adjectives and adjectival metaphors in VUAMC. This again reinforces the challenges in relying on annotated datasets for metaphor generation tasks – even the largest ones are not nearly as large as we would like given the prevalence of metaphor in language use generally.

5. Conclusion

With this paper I have given a brief overview of the state of research in metaphor generation today. It is an exciting field where developments are occurring quickly, as can be seen in the fact that the three papers I have summarised on the subject were all published within about the last two years as of writing. There does however remain much to be explored, and many of the challenges still come back to the issues of effective metaphor identification and the reliance on the small number of annotated datasets that exist. Suppose for example that an outstanding binary model was devised that could do effective word-level predictions of whether a unit was metaphorical or literal. This would of course be useful but I suspect it would lead only to a brittle generation model that could simply replace a literal part of speech with a metaphorical one without regard for all of the nuances in metaphor use as codified in a system like MIPVU. Knowing whether a word or phrase is metaphorical is not enough – it would be better to know whether it was a direct metaphor or an indirect metaphor, implicit or personifying, dead or alive.

An ideal scenario may be that an unsupervised metaphor identification model is developed that could be able to capture close to as much nuance as MIPVU does in terms of metaphor categorisation. I am not sure if this is feasible given current deep learning-based language models as they would need to be able to infer more complex linguistic relationships than they do at present.

Based on what I had learned of the domain of metaphor generation, I have here tried to contribute to the field by first creating an adjective metaphor corpus based on VUAMC, then by using this to fine-tune a BERT model to classify unseen adjectives as metaphorical or literal. The results show promise: the model is correct three out of four times when it predicts that an adjective is metaphorical, while it manages to identify about half of the metaphorical adjectives in the test set.

It would be interesting to follow the same steps that MERMAID did for verbs by using this adjective classifier model on the Gutenberg Poetry Corpus to create parallel training data before going on to generate adjectival metaphors. In order to investigate the feasibility of broader literal-to-metaphorical replacement, I would then like to see if I could combine the verb and adjective models to gauge whether both could be applied simultaneously.

Bibliography

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Beata Beigman Klebanov, Chee Wee (Ben) Leong, and Michael Flor. 2018. A corpus of non-native written English annotated for metaphor. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 86–91, New Orleans, Louisiana. Association for Computational Linguistics.

Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan and Nanyun Peng. 2021. MERMAID: Metaphor Generation with Symbolism and Discriminative Coding. arXiv:2103.06779v2

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

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

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, pages 1222–1231, Melbourne, Australia. Association for Computational Linguistics.

Rui Mao, Chenghua Lin, and Frank Guerin. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3888–3898.

Iain McGilchrist. 2009. The Master and His Emissary: The Divided Brain and the Making of the Western World. New Haven: Yale University Press.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In Procee

dings of the Fifth Joint Conference on Lexical and Computational Semantics, pages 23–33, Berlin, Germany. Association for Computational Linguistics

Andrzej Pawelec. 2007. The Death of Metaphor. Studia Linguistica. Universitatis Iagellonicae Cracoviensis 123: 117-122

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1314–1324. Association for Computational Linguistics.

Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. Metaphor and Symbol, 22 (1), 1–39.

Princeton University. 2010. *About WordNet*. WordNet.

Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. arXiv preprint arXiv:1709.00575.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 160–170.

Gerard Steen, Aletta Dorst, J. Berenike Herrman, Anna Kaal, Tina Krennmayr, and Tryntje Pasma. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*, volume 14. John Benjamins Publishing.

Kevin Stowe, Leonardo Ribeiro, and Iryna Gurevych. 2020. Generating Metaphoric Paraphrases. arXiv:2002.12854v1

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to Fine-Tune BERT for Text Classification? arXiv:1905.05583

Hao Yaru, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and Understanding the Effectiveness of BERT. arXiv:1908.05620