Ranking Creative Language Characteristics in Small Data Scenarios

Julia Siekiera[1], Marius Köppel[2], Edwin Simpson[3,4], Kevin Stowe[4], Iryna Gurevych[4], Stefan Kramer[1]

[1]Dept. of Computer Science and [2]Institute for Nuclear Physics, Johannes Gutenberg-Universität Mainz,
{siekiera,mkoeppel}@uni-mainz.de, kramer@informatik.uni-mainz.de,
[3]Dept. of Computer Science, University of Bristol, edwin.simpson@bris.ac.uk,
[4]Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt
https://www.informatik.tu-darmstadt.de/ukp

Abstract

The ability to rank creative natural language provides an important general tool for downstream language understanding and generation. However, current deep ranking models require substantial amounts of labeled data that are difficult and expensive to obtain for new domains, languages and creative characteristics. A recent neural approach, DirectRanker, reduces the amount of training data needed but has not previously been used to rank creative text. We therefore adapt DirectRanker to provide a new deep model for ranking creative language with small numbers of training instances, and compare it with a Bayesian approach, Gaussian process preference learning (GPPL), which was previously shown to work well with sparse data. Our experiments with short creative language texts show the effectiveness of DirectRanker even with small training datasets. Combining DirectRanker with GPPL outperforms the previous state of the art on humor and metaphor novelty tasks, increasing Spearman's ρ by 25% and 29% on average. Furthermore, we provide a possible application to validate jokes in the process of creativity generation.

Introduction

To process or evaluate creative language, natural language processing systems need to recognise figurative and humorous expressions, so that they do not interpret jokes or metaphors literally, and can gauge different aspects of creativity. The simple binary recognition of figurative or humorous language is not sufficient, as different examples require varying degrees of creativity, and hence different kinds of processing. Consider the following metaphors:

- This view has been **attacked** on the grounds that it...
- She **attacked** the sandwiches like a starving bear.

In both examples, the verb 'attack' strays from the literal meaning of a military offensive, but the first usage is very conventional, while the second appears much more novel and metaphoric. Other properties of creative language, such as humor, have similar gradations, which motivates methods for ranking sentences according to these properties.

The process of creativity is highly complex and nontrivial to automate, but creative writers may benefit from automated tools. As the comedy writer Charlie Skelton said: *to begin with, we must ask: what is the metric for a "successful" joke? ...Is it one that makes the most people laugh, or the right people laugh, or its own creator laugh?* (Skelton 2021), the success of a joke is strongly cultural-based. He also describes the creative process of a professional comedy writer creating a joke: *a joke can be judged, and just as many checkboxes to tick on its journey from the writer's mind to the audience's ears.* In the setup of the Componential Model of Creativity (Press 2017; Press 2011) this can be seen as the response validation and communication step. To help with this step, a ranking model could provide an automated evaluation method to help a comedy writer answer the question: "Is this joke the one that makes the most people laugh?". Ranking models trained with data annotated from various cultural backgrounds could also give insights into how they may perceive different jokes.

To obtain training data for a ranking model, annotators could assign scores to individual examples, but inconsistencies can arise between annotators and across the labels of a single annotator over time. We therefore turn to pairwise comparisons between examples, which simplify the annotators' task and avoid the need to calibrate their scores. A ranker can then derive the entire ranking from pairwise labels. Considering the cost of annotating data for different domains, languages and aspects of creativity, we need a ranker that can be trained on datasets with a small number of examples and sparse pairwise labels. For ranking creative language, Simpson and others (2019) adopted *Gaussian process preference learning (GPPL)*, a Bayesian approach that uses word embeddings and linguistic features and can cope with sparse and noisy pairwise labels. However, it is a shallow model that relies on predetermined features to represent each example.

In contrast, neural network architectures can learn representations directly from pairwise comparisons, but demand a higher quantity of training labels. A recent method, *DirectRanker* (Köppel and others 2019) improves label efficiency for document ranking by fulfilling the requirements of a total quasiorder in the model architecture, which results in faster convergence than other neural network ranking approaches, as this order does not have to be learned. This paper adapts DirectRanker to text ranking for the first time, setting a new state of the art for humor and metaphor novelty, showing that even with limited data, text ranking can benefit from deep representation learning. Our experiments show that combining Bayesian and neural approaches using stacking can improve further ranking quality. While we find a clear benefit to BERT embeddings (Devlin and

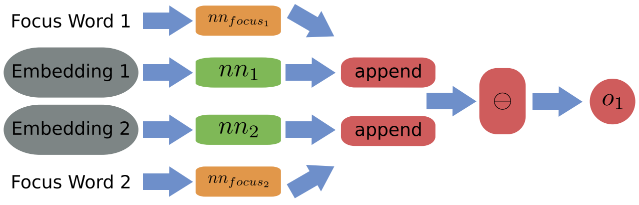

Figure 1: The adapted DirectRanker architecture. Embeddings are fed into the parameter sharing networks nn_1 and nn_2 to generate representations (feature part). For datasets containing focus word information, we add parameter sharing networks nn_{focus_1} and nn_{focus_2}. The (appended) representations are subtracted and fed into the ranking part (in red) with output neuron o_1 that has no bias and $\tanh$ as activation.

others 2018) for humor, current embedding methods have difficulty in modelling metaphors. To support the evaluation of creative content, we make our software available at `https://zenodo.org/record/6275546`.

Related Work

Algorithms solving the ranking problem can be divided into three categories. *Pointwise* rankers assign a score to each document (Cooper and others 1992). *Pairwise* models predict which document is more relevant out of two for a given query (Köppel and others 2019). *Listwise* algorithms optimise a loss function that considers the whole ordered list of documents (Cao and others 2007). Previous research on document ranking combined BERT (Devlin and others 2018) with different learning-to-rank methods of all three categories. While Han and others (2020) and Qiao and others (2019) embed concatenated queries and documents with BERT and fine-tune ranking performance using an arbitrary artificial neural network ranker, Nogueira and others (2019) introduce a multi stage pipeline containing a pointwise and a pairwise BERT ranker to trade off ranking quality against latency. However, these approaches are evaluated neither for small training data scenarios nor on the difficult task of creative language and lack the label-efficient learning property that DirectRanker introduces. In the past, DirectRanker was used for ranking multilingual BERT models (Chen and Ritter 2020), but the approach ranks the models themselves rather than text documents, which we address here.

DirectRanker for Text Ranking

DirectRanker, shown in Figure 1, consists of a *feature* part, which learns a low-dimensional latent representation of the input documents, and a *ranking* part, which receives the latent representations for a pair of examples and predicts a pairwise label. The ranking part is used to train the model from pairwise labels, but can also be used after training to predict the degree of creativity for any arbitrary text.

To adjust the DirectRanker to text ranking, we include dropout layers and batch normalization in the networks nn_1 and nn_2 to reduce overfitting. For some creative language tasks such as metaphor novelty prediction, the aim is to evaluate the use of a specific word or phrase within a larger context. Hence we need to represent both the word or phrase (henceforth the *focus word*) and the sentence that contains it. During initial experiments, we found that transforming the sentence and focus word together in nn_1 and nn_2 leads to unequal weighting of both information sources in the feature part, as the two feature vectors differ in length and in their most extreme values. We therefore add the networks nn_{focus_1} and nn_{focus_2} to the feature part to process the focus words separately from their context. This facilitates training as the model is able to weight the compressed sentence and focus word information in the less complex ranking part. The results of both the sentence network and the focus word network are concatenated and passed to the ranking part.

The ranking function is given by $o_1(x_1, x_2) = \tau\left(w\left(\frac{(u_1,u_{f_1})-(u_2,u_{f_2})}{2}\right)\right)$, where $u_1 = nn_1(x_1)$ and $u_2 = nn_2(x_2)$ compress the input feature vectors x_1 and x_2 to latent representations u_1 and u_2, u_{f_1} and u_{f_2} are latent representations for the focus words computed by nn_{focus_1} and nn_{focus_2}, w represents the multilayer perceptron ranking weights for the last neuron and τ is an antisymmetric sign conserving activation. The loss function remains the same as in the original DirectRanker paper: $L_{\text{rank}}(\Delta y, x_1, x_2) = (\Delta y - o_1(x_1, x_2))^2$, where Δy is the gold pairwise label in the training set. Beside the changes of the feature part, we included the possibility to change the ranking part to a Gaussian Process layer using a Matérn kernel, enabling a direct combination with the ideas of the GPPL model. Therefore, the original ranking function can be replaced with $p(x_1 \succ x_2) = \Phi\left(\frac{u_1-u_2}{\sqrt{2\sigma^2}}\right)$ for the ranking part, where $x_1 \succ x_2$ indicates that instance x_1 was labeled as preferred to x_2, Φ is the probit function, and σ^2 is a variance parameter.

Text Representation We investigate three text representations. First we choose mean *word2vec* embeddings (MWE) trained on part of Google News (Mikolov and others 2013) to directly compare the findings of Simpson and others (2019) with the DirectRanker. However, *word2vec* embeddings have the disadvantage that they assign a single, fixed representation for each word, even though it may take on different meanings in different contexts, particularly with regard to creative language. To address this, we fine-tune BERT with DirectRanker to produced contextual word embeddings, and again take the mean to represent the whole sentence. To better capture the meaning of a whole sentence, we apply sentence transformers (Reimers and Gurevych 2019) to generate sentence embeddings (SEs). In contrast to MWEs, sentence transformers learn how to compose individual contextual word embeddings and assign sentences with similar meanings close representations in the vector space.

Datasets

We explore GPPL and DirectRanker on two datasets including different types of creative language. The *humor* dataset (Simpson and others 2019) is an extension of Miller and others (2017), which contains 4030 samples with various degrees of humorousness, with an average sentence length of 11 words. The humorous examples can be grouped into

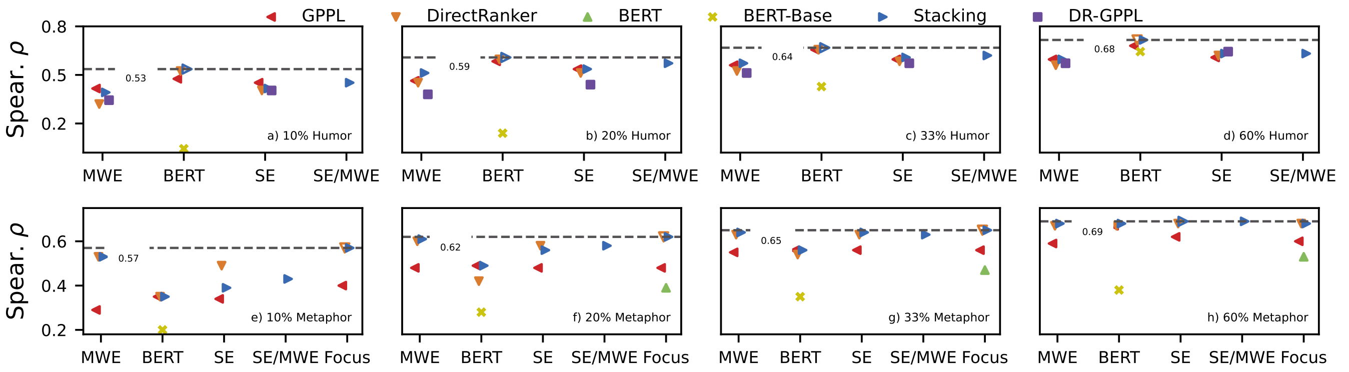

Figure 2: Mean results with different training set sizes. Humor results are shown in the top row, Metaphor results in the bottom row. Each plot shows different splits of the datasets. The different embeddings used by the models are marked on the x-axis. The Stacking method has the possibility to use both SE and MWE. We show Spearman's ρ against the gold score. For better visibility we excluded the result for BERT with focus word embeddings for 10% Metaphor (Spearman's ρ of -0.03) and we added different x-axis offset for the models. A detailed table of the displayed data can be found in Table 1.

homographic and heterographic puns containing purely verbal humor while the non-humorous section contains proverbs and aphorisms. The *metaphor* dataset (Do Dinh and others 2018) contains 72816 examples from the VU Amsterdam Metaphor Corpus (Steen and others 2010) that have been annotated for metaphor novelty, including metaphors in four genres: news, fiction, conversation transcripts, and academic texts. Each example consists of a sentence and a labeled focus word. Both datasets were labeled using crowdsourcing. For the humor dataset, every instance was selected for 14 random pairwise comparisons and each pair was labeled by 5 different annotators. For the metaphor dataset, each instance was included in 6 random tuples, each containing 4 instances. Each tuple was labeled by 3 annotators, who selected the most novel and most conventionalised examples from each tuple. We did not survey the background of the annotators, other than to check that they are proficient in English. We generate pairwise labels between the most novel and most conventionalised samples in each tuple, following Simpson and others (2019). The resulting pairwise comparisons are labeled 1.55 times on average and each instance is present in 8.6 pairs on average.

Experimental Setup

We evaluate our experiments using 4 internal folds for finding the best hyperparameters and 3 external folds for evaluation. To examine the ranking performance on sparse data, we also experiment with artificially reducing training set sizes. For this purpose, we randomly select 60%, 33%, 20% and 10% of the example IDs and train on only the pairs where both examples are in our selection. The remaining samples are used in the test set to counteract the model variation for smaller training sets. The DirectRanker feature part is a 4-layer fully-connected network with 2k, 500, 64 and 7 neurons in each layer. To evaluate the effect of the Gaussian Process layer in the ranking part, we run the experiments on the humor dataset two times, once with and once without the Gaussian Process layer. Code from Simpson and others (2019) was used to train and obtain predictions from GPPL using a Matérn $\frac{3}{2}$ kernel. To improve the overall ranking performance, we combine the predictions of GPPL and the DirectRanker with stacking, using a linear regression model to weight the predictions of the two models. To generate SEs, we use the pretrained 'bert-base-nli-stsb-mean-tokens' model. We use 'bert-base-cased' for fine-tuning BERT with the DirectRanker and reuse the resulting embeddings with GPPL. The methods are evaluated by computing the linear rank correlation between the prediction and the gold standard with Spearman's ρ.

Results of Method Comparison

The results are shown in Figure 2. As a baseline, we include BERT regression models fine-tuned directly on the rankings in the training sets (indicated in Figure 2 with a gold x as BERT-Base). For the metaphor data, we extend the BERT regression model to incorporate the *word2vec* embedding of the focus word as a further input (indicated in Figure 2 with the green △). In all cases, both BERT regression and the state-of-the-art GPPL are out-performed by either DirectRanker and Stacking. We highlighted the best model by adding a horizontal line annotated with the Spearman's ρ value and removing the filling. The standard deviation ranges from 0.016 for 60% to 0.038 for 10% on Humor and from 0.006 for 60% to 0.043 for 10% on Metaphor dataset.

On the humor dataset, the BERT baseline performs well in the 60% case as it is able to classify the less relevant documents better. However, the baseline is not suitable for scenarios with less data, in which the pairwise models achieve significantly better results. On Humor, GPPL outperforms the DirectRanker on almost all training set sizes and text representations except for BERT and 60%. The 60% case with SE was the only one where the Gaussian Process layer in the ranking part (DR-GPPL) outperforms the normal DirectRanker approach. Both GPPL and the DirectRanker benefit most from BERT, but the DirectRanker particularly benefits from the pretrained BERT with small training sets. By combining GPPL and DirectRanker, both with BERT, stack-

	Humor				Metaphor			
	60%	33%	20%	10%	60%	33%	20%	10%
Bert Baseline	0.62	0.44	0.20	0.12	0.38	0.35	0.28	0.20
Bert + Focus Word	-				0.53	0.47	0.39	-0.03
GPPL MWE	0.54	0.53	0.47	0.41	0.58	0.55	0.51	0.35
DirectRanker MWE	0.54	0.50	0.44	0.30	0.64	0.60	0.52	0.37
DR-GPPL SE	0.62	0.56	0.45	0.42	-			
DR-GPPL MWE	0.56	0.51	0.40	0.37	-			
Stacking MWE/MWE	0.58	0.56	0.51	0.41	0.68	0.64	0.61	0.53
Stacking BERT/BERT	0.68	0.64	0.59	0.53	0.68	0.56	0.49	0.35
Stacking SE/SE	0.61	0.59	0.53	0.43	0.69	0.64	0.56	0.39
Stacking SE/MWE	0.61	0.60	0.56	0.46	0.69	0.63	0.58	0.43
GPPL ∅ MWE	0.58	0.55	0.47	0.43	0.59	0.55	0.48	0.29
GPPL ∅ BERT	0.65	0.63	0.57	0.48	0.67	0.56	0.49	0.35
GPPL ∅ SE	0.59	0.58	0.53	0.46	0.62	0.56	0.48	0.34
DirectRanker ∅ MWE	0.55	0.52	0.46	0.35	0.67	0.63	0.60	0.53
DirectRanker ∅ BERT	0.68	0.63	0.58	0.52	0.67	0.54	0.42	0.35
DirectRanker ∅ SE	0.62	0.57	0.51	0.42	0.68	0.63	0.58	0.49
Stacking Focus Word	-				0.68	0.65	0.62	0.57
GPPL ∅ Focus Word	-				0.60	0.56	0.48	0.40
DirectRanker ∅ Focus Word	-				0.68	0.65	0.62	0.57

Table 1: Mean results with different training set sizes on the two datasets. We show Spearman's ρ against the gold score. The ∅ indicates that the model's mean score of the 4-fold cross-validation ensemble is evaluated (see the end of Section Stacking. For stacking we first name the embeddings used for GPPL and then for DirectRanker.

ing is able to improve the individual performances across all training set sizes. A similar improvement is shown for other stacking setups, for example with GPPL on SEs and DirectRanker on MWEs (Stacking SE/MWE).

On the metaphor dataset the models' behavior changes. The BERT baseline is not able to reach competitive results in any training scenario and the BERT embeddings do not consistently improve over other embeddings, supporting previous results where BERT underperforms on metaphor tasks (Mao, Lin, and Guerin 2019). DirectRanker outperforms GPPL on most combinations, especially on smaller training sets. For instance, DirectRanker outperforms GPPL with MWE and SE, including for 10% and 20% datasets, showing its suitability for small datasets. In most settings, stacking maintains or slightly exceeds the ranking performance in each combination. In the 20% and 10% case, stacking falls below the maximum individual performance on the SEs as GPPL overfits on the validation set. This might be an effect of learning with SEs on a small training and validation set so that they are not representative of the test set. For metaphor novelty, the models trained on only the *word2vec* focus word embedding outperform those that are also trained with sentence representations with 33% - 10% training data. Furthermore, neither GPPL nor DirectRanker are able to extract much useful information from the sentences alone. With SEs in the 60% case, DirectRanker and GPPL reach a Spearman's ρ of 0.64 and 0.58, respectively. While this may reflect a limitation of the sentence representations, it is also possible that the annotators who produced the gold standard fixated too strongly on the focus words.

Conclusion

In this work we investigated a pairwise ranking approach for creative language based on adapting a recent neural architecture, DirectRanker, that can learn efficiently from small training sets. We combined it with a Bayesian model, GPPL, and evaluated the behavior of all models on the tasks of predicting humorousness and metaphor novelty with different text representations. Despite the expectation that neural networks suffer from overfitting on small datasets, DirectRanker was able keep up with or even improve on GPPL. The proposed stacking approach clearly outperforms state-of-the-art results and is a powerful tool for language tasks with a limited number of training documents. On the humor dataset we showed a substantial ranking improvement over pretrained embeddings by fine-tuning BERT with the DirectRanker architecture. Due to the heavy reliance on the focus word information, this was less effective for the metaphor dataset, where the best results were achieved using only the focus words' *word2vec* embeddings. Resent work showed that using the integration of constructional semantics and conceptual metaphor showed better generalizations across metaphoric and non-metaphoric language (Sullivan 2016). While others provided alternatives to the representation of contextual information, such as the cultural context (Cabezas-García and Reimerink 2022). Using these different approaches could be beneficial for providing better representations for metaphor.

A possible application, in the context of joke generation, is the evaluation of creative content. The ranked sentences can help to evaluate jokes and quantify whether they are funny for a majority of people. Nevertheless, this method comes with limitations since it is not aware of any context the joke was made. To see this, consider this cherry-picked example of a joke which was ranked high: "I do a lot of **spreadsheets** in the office so you can say I'm **excelling** at work." While the model was able to characterize that this sentence is a pun, the context in which this joke is funny was never present. To understand the joke, one needs to know that working in the office often means working with Excel. This knowledge is not present to everyone and would only be understand in the current time, when Microsoft products are widely used. In further work the model can be used to compare the views of people from different cultural backgrounds on particular kinds of humour or metaphor. Our results show that further work is required to develop better representations of context, particularly for evaluating metaphors. Our analysis considered a relatively narrow type of humor – puns – which we will expand upon in future work. Another important direction is to develop suitable representations for longer texts where sentence embeddings may be less representative of creative language characteristics.

Acknowledments

We would like to thank Pia Snella for proof(read)ing the social implications of our model.

Parts of this research were conducted using the supercomputer MOGON 2 and/or advisory services offered by Johannes Gutenberg-University Mainz (hpc.uni-mainz.de), which is a member of the AHRP (Alliance for High Performance Computing in Rhineland-Palatinate, www.ahrp.info) and the Gauss Alliance e.V.

The authors gratefully acknowledge the computing time granted on the supercomputer MOGON 2 at Johannes Gutenberg-University Mainz (hpc.uni-mainz.de).

This project was partly funded by the project RMU (Rhine-Main Universities) Network for Deep Continuous-Discrete Machine Learning (DeCoDeML) of RMU-Initiativfonds and by the "Data Analytics for the Humanities" grant by the Hessian Ministry of Higher Education, Research, Science and the Arts.

Author Contributions

JS developed the DirectRanker for text ranking. MK adapted the Gaussian Process layer for the use with the DirectRanker. The experiment design was done by JS, MK, ES and KS. JS performed and analyzed the experiments shown in Table 1 and Figure 2 with the help of MK. The manuscript was written by JS, MK, ES and KS. IG and SK supervised the study and reviewed the manuscript.

References

[Cabezas-García and Reimerink 2022] Cabezas-García, M., and Reimerink, A. 2022. Cultural context and multimodal knowledge representation: Seeing the forest for the trees. In *Frontiers in Psychology*.

[Cao and others 2007] Cao, Z., et al. 2007. Learning to rank: From pairwise approach to listwise approach. In *ICML*.

[Chen and Ritter 2020] Chen, Y., and Ritter, A. 2020. Model selection for cross-lingual transfer using a learned scoring function. In *arXiv:2010.06127*.

[Cooper and others 1992] Cooper, W. S., et al. 1992. Probabilistic retrieval based on staged logistic regression. In *ACM SIGIR*.

[Devlin and others 2018] Devlin, J., et al. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. In *arXiv:1810.04805*.

[Do Dinh and others 2018] Do Dinh, E.-L., et al. 2018. Weeding out conventionalized metaphors: A corpus of novel metaphor annotations. In *EMNLP*.

[Han and others 2020] Han, S., et al. 2020. Learning-to-rank with BERT in TF-Ranking. In *arXiv:2004.08476*.

[Köppel and others 2019] Köppel, M., et al. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. In *ECML*.

[Mao, Lin, and Guerin 2019] Mao, R.; Lin, C.; and Guerin, F. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *ACL*.

[Mikolov and others 2013] Mikolov, T., et al. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

[Miller and others 2017] Miller, T., et al. 2017. Detection and interpretation of English puns. In *SemEval*.

[Nogueira and others 2019] Nogueira, R., et al. 2019. Multi-stage document ranking with BERT. In *arXiv:1910.14424*.

[Press 2011] Press, A. 2011. Research and methods. In *Encyclopedia of Creativity (Second Edition)*.

[Press 2017] Press, A. 2017. Understanding creativity in the performing arts. In *Creativity and the Performing Artist*.

[Qiao and others 2019] Qiao, Y., et al. 2019. Understanding the behaviors of BERT in ranking. In *arXiv:1904.07531*.

[Reimers and Gurevych 2019] Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Simpson and others 2019] Simpson, E., et al. 2019. Predicting humorousness and metaphor novelty with Gaussian process preference learning. In *ACL*.

[Skelton 2021] Skelton, C. 2021. Comedy by numbers — a comedy-writer's thoughts on algorithmic approaches to humour. In *ICCC*.

[Steen and others 2010] Steen, G., et al. 2010. A method for linguistic metaphor identification. from MIP to MIPVU. In *CELCR*.

[Sullivan 2016] Sullivan, K. 2016. Integrating constructional semantics and conceptual metaphor. In *Constructions and Frames*.