Metaphor Detection for Low Resource Languages: From Zero-Shot to Few-Shot Learning in Middle High German

Felix Schneider[1], **Sven Sickert**[1], **Phillip Brandes**[2], **Sophie Marshall**[2], **Joachim Denzler**[1]
[1] Computer Vision Group, [2] Institut für Germanistische Literaturwissenschaft
Friedrich Schiller University Jena
Jena, Germany
firstname.lastname@uni-jena.de

Abstract

In this work, we present a novel unsupervised method for adjective-noun metaphor detection on low resource languages. We propose two new approaches: First, a way of artificially generating metaphor training examples and second, a novel way to find metaphors relying only on word embeddings. The latter enables application for low resource languages. Our method is based on a transformation of word embedding vectors into another vector space, in which the distance between the adjective word vector and the noun word vector represents the metaphoricity of the word pair. We train this method in a zero-shot pseudo-supervised manner by generating artificial metaphor examples and show that our approach can be used to generate a metaphor dataset with low annotation cost. It can then be used to finetune the system in a few-shot manner. In our experiments we show the capabilities of the method in its unsupervised and in its supervised version. Additionally, we test it against a comparable unsupervised baseline method and a supervised variation of it.

Keywords: metaphor detection, low resource language, middle high german, zero-shot learning, few-shot learning

1. Introduction

The automatic detection of metaphors is a useful tool for literary studies. While many recent supervised approaches for common languages like English exist, those methods rely on large pretrained models like BERT (**?**) transformers and on labeled metaphor datasets, as can be seen in the shared task by **?**). Both can not be obtained for low resource languages like Middle High German (MHG), which is an older form of German spoken between around 1050 AD and 1350 AD. To enable metaphor detection in such cases we propose a novel unsupervised zero-shot approach based only on simple word embeddings. In our approach, a feedforward neural network transforms the word embeddings of adjective-noun metaphor word pairs into another vector space. This space has the property that common literal word pairs are located near each other while metaphoric word pairs have a large cosine distance between them. This distance can serve as a measure of metaphoricity. We are especially interested in *intentional* metaphors, which are actively used by the authors, and not in so-called *dead* metaphors, wich have experienced a shift in meaning to also include their metaphorical meaning in their base meaning (e.g. leg of a chair), while also recognizing that there exist combinations which may not unambiguously belong to one of those classes.

A metaphor, as a semantic figure of speech, is a way of referring to one concept by mentioning another (**?**). An example for this would be the phrase *the car drinks gasoline* (**?**), where the word *drinks* from the domain of food consumption is applied to word *car* from the domains of transportation and machines. It carries over its base meaning of consumption of liquids, so that the reader understands that the car consumes fuel. Another example would be the phrase *a sweet thought*. Here the word *sweet* from the domain of taste is applied to the word *thought*. While in its base meaning only physical objects can be sweet, the reader understands by their context knowledge and world knowledge that a sweet taste is considered pleasant and thus the aforementioned phrase means a pleasant thought.

In this work, we concentrate on adjective-noun pattern like *sweet thought*, *raw emotion*, or *clear answer*. With the knowledge of syntactical dependencies also more complex forms can be analyzed. However, we want to limit our approach to methods also applicable to low resource languages like MHG, where no syntax parsing is available. Thus, we assume that only part-of-speech tags and token-based word embeddings like word2vec (**?**) or fastText (**?**) are obtainable. We do not rely on methods requiring large amounts of training data like transformer models or syntax parsers.

There are different ways to define adjective-noun metaphors to operationalize the search for them. An overview of approaches can be seen in the work of **?**). One possibility is to define metaphors as a violation of the selectional preference of a word (**?**; **?**). The approach we focus on defines the adjective that commonly occur together with a noun as their selection preference. When an adjective that does not typically appear together with the noun emerges, this anomaly is called a selection preference violation. This implies that an adjective from another source domain is used to describe something from the target domain of the noun. It fits our definition of a metaphor. Since our approach should also be applicable to new languages without an existing labeled metaphor dataset in that language, we need to develop an *unsupervised* approach. In Sec-

tion 3. we explain how to derive such a method from a supervised method.

2. Related Work

The most recent approaches for metaphor detection are based on supervised learning and transformer models such as MIss RoBERTa WiLDe (**?**), MelBERT (**?**), and DeepMet (**?**) . Those models require to be pretrained on a very large corpus with billions of tokens. However, there do not exist corpora of sufficient size to pretrain large language models on for every language. If we want to search for metaphors in low resource languages like MHG, using such a large pretrained language model is not possible. Additionally, there may be no training dataset for supervised training available to finetune the model on.

Other approaches like (**?**) use supersense taxonomies like GermaNet (**?**; **?**) comparable to the English WordNet (**?**). They deliver information about the domain that certain words belong to. However, those external sources of information are not present for low resource languages like MHG. In an earlier unsupervised approach, the authors of (**?**) used grammatical relations between words as the basis for a clustering approach based on hierarchical graph factorization. For this approach syntax parsing is necessary, as well. The authors of (**?**) propose an unsupervised metaphor detection system based on topic modeling. In comparison, they do not search for adjective-noun pairs but instead for single words with metaphorical meaning inside a sentence.

However, there are also unsupervised approaches that do not rely on big pretrained transformer models. Our *baseline* (**?**) clusters adjective-noun pairs using the kmeans algorithm. To cluster the data, six different features are used: (1) abstractness rating of the adjective; (2) abstractness rating of the noun; (3) difference between the abstractness ratings; (4) cosine similarity of the word embeddings of the noun; (5) edit distance from the adjective to the noun, normalized by the number of characters in the adjective; (6) edit distance from the noun to the adjective, normalized by the number of characters in the noun. Clusters are then interpreted as metaphors or non-metaphors. This approach uses information that may not be present in low resource languages (the abstractness rating). However, we consider this a comparable baseline approach to our work. Due to its unsupervised nature, it can also be used on languages without an existing metaphor dataset.

3. Method

Our contribution consists of two parts: First, we propose a feedforward neural network that maximizes the cosine distance between the word vectors of an adjective-noun word pair for metaphors and minimizes the distance otherwise. Second, a way to train this model in a zero-shot setting without any metaphor examples. It also covers a step to finetune the system on human annotated metaphors previously proposed by the unsupervised system.

3.1. Metaphor Ranking

The basic idea of our novel approach is to transform the word embeddings of the adjective and the noun into another vector space, where the distance between words is based on their metaphoricity instead of their co-occurence. The cosine distance between the transformed vectors is small if the word pair is meant literally and large if the word pair has a metaphorical function. We assume, that words which occur often next to each other should have a low distance by the nature of the word embeddings. At the same time, unusual combinations like metaphors should have a higher distance. However, this is not guaranteed, especially with low resource data. As an extreme example, if the whole available corpus consists of poetry, words may be used in a metaphorical context more often than with their literal meaning. Additionally, while hapax legomena in large corpora normally comprise niche expressions, in a low resource language corpus also central words may be hapax legomena.

Our approach thus transforms the word embeddings into a space, where this higher distance between metaphorical words is explicitly encouraged. To transform the word embeddings into the metaphoricity vector space, we use a simple feedforward network N. The network for the transformation of the word embedding e_a of the adjective is the same as for the word embedding e_n of the noun, resulting in their transformed vectors t_a and t_n. This reduces the number of parameters that need to be learned. We then determine the metaphoricity m of the word pair by computing the cosine distance Δ_{cos} of the transformed vectors, as seen in Equation 1.

$$m = \Delta_{cos}(t_a, t_n), t_a = N(e_a), t_n = N(e_n) \quad (1)$$

The cosine embedding function (**?**) is used as a training loss. It maximizes the cosine distance between the transformed vectors if the word pair has a metaphorical meaning and minimizes the distance if the word pair has a literal meaning. Hence, the cosine distance of the transformed vectors then represents the metaphoricity of a word pair and can be used to rank all possible metaphor candidates.

3.2. Unsupervised Zero-Shot Training

As a goal, we also want to apply this method to low resource languages like MHG where we do not have a labeled metaphor dataset. This renders supervised training impossible. To mitigate this, we assume the number of metaphorical adjectives in a text to be low enough to make a high amount of adjective-noun pairs in a text good examples for non-metaphors. Based on this assumption, we generate artificial metaphor examples by using the idea of selectional preference viola-

tion. We create artificial metaphors by generating random adjective-noun pairs and label those as metaphor examples. While this may not result in semantically useful metaphors, it still satisfies the idea of selectional preference violation to initially train the neural nework. It enables the classifier to distinguish between normal and anomalous word pairs. Afterwards, the trained model can be used to extract real metaphors from the corpus, annotate those and finetune the model.

3.3. Few-Shot Finetuning

With the above mentioned idea, we get a classifier to rank the metaphoricity of adjective-noun pairs using no labeled training data. While the created classifier is not yet specifically tuned for real metaphors, we use it to evaluate how uncommon a word combination is. In contrast to using probability tables of word combinations or similar approaches, our word embedding based approach can also rank word pairs which have not been seen in the training data based on their semantic similarity encoded in the embeddings. Especially in low resource languages with small and non-representative corpora, the infrequent co-occurrence of words may not be sufficient to deduce their metaphoricity.

Our model can thus be refined with a human-in-the-loop bootstrapping approach. Using the zero-shot classifier, we can rank all the adjective-noun pairs in the training corpus by their estimated metaphoricity. An expert can then annotate metaphor candidates based on the ranking to generate a metaphor dataset without the need to annotate the whole text. As our strategy we choose to annotate the top 100 ranked word pairs, the bottom 50 ranked pairs and 50 random examples in every step. We repeat this in an iterative manner, generating metaphor examples of increasing quality with every annotation step. Thus, we create both a metaphor detection model and a dataset without the need to annotate whole corpora.

4. Experiments

To evaluate our embedding approach as well as our unsupervised labeling approach, we conducted several experiments. For reproducability, we make our code publicly available [1]. Since we want to emulate the search for metaphors in low resource languages, we do not use all features that are possible in the German language. Syntax trees, external knowledge bases like GermaNet and large pretrained models like BERT are excluded.

4.1. Data and setup

As a corpus for the German case study to extract non-metaphors in an unsupervised manner, we used the GerDraCor (Fischer et al., 2019) corpus. For the case study on the low resource language MHG, we used the Referenzkorpus Mittelhochdeutsch (Klein et al., 2016) to train fastText (**?**) word embeddings. This corpus contains about 2,000,000 words. The model was trained using the *skipgram* approach with 1000 epochs and a learning rate of 0.01 on 8 threads with an embedding vector size of 100. A word vector for every word in the corpus was generated, resulting in 56060 vectors. We took 22 texts from the Mittelhochdeutsche Begriffsdatenbank (Zeppezauer-Wachauer, 2022) to analyze our approach on this language. The CLTK (**?**) package was used to normalize the character representation of the MHG texts and to generate PoS tags. We extracted PoS tags, tokens, and word embeddings for the German data using the spaCy (**?**) package.

As annotated metaphor dataset we used the German version (**?**) of the TSV metaphor dataset. Additionally, we used their annotated metaphor dataset from German poetry. However, their approach used features based on GermaNet, a supersense taxonomy which can not be assumed to exist for low resource languages. Hence, we did not compare our method to theirs. For the TSV dataset the training set comprised 546 metaphors and 603 non-metaphors, the test set comprised 65 metaphors and 77 non-metaphors, while for the poems dataset the training set comprised 100 metaphors and 487 non-metaphors, the test set comprised 98 metaphors and 280 non-metaphors. Our neural network had an input size of 300 for German and 100 for MHG, two hidden layers of size 300 and an output layer of size 100. ReLU was used as an activation function for the hidden layers.

4.2. Baseline

The main advantage of our appraoch is that it uses only POS tags as additional information, while the word embeddings can be learned from a corpus. Since even most very simple methods for metaphor detection use additional information like syntax trees, it is not easy to find a suitable baseline to compare to our approach. As baseline we used the methods explained in Section 2. Since the abstractness features are not present in low resource languages, we also conducted an experiment without these features. The remaining features are the cosine similarity of the word embeddings of the noun, the edit distance from the adjective to the noun, normalized by the number of characters in the adjective, and the edit distance from the noun to the adjective, normalized by the number of characters in the noun. While our baseline method is primarily an unsupervised approach, our approach can also be used in a supervised manner. For a fair comparison with our supervised approach, we also used the baseline features with a kernel SVM in a supervised manner.

4.3. Supervised metaphor retrieval

In the most simple case we have a dataset consisting of word pairs which are either labeled as a metaphor or as non-metaphor. Given these labels, our approach can be used without any modification. For our baseline, we trained a kernel SVM with radial basis function (RBF) kernel with the features of the otherwise unsupervised baseline by (**?**). As hyperparameters for

[1] https://github.com/cvjena/metaphor-detector

method	TSV	poems
supervised (ours)	0.90	0.82
SVM baseline features (+abst)	0.92	0.77
SVM baseline features	0.67	0.75
zero-shot GerDraCor (ours)	0.70	0.74
zero-shot (ours)	0.57	0.77
baseline (+abst)	0.86	0.76
baseline	0.57	0.79

Table 1: Results of two different experiments: numbers are the average precision, which is the area under the precision-recall-curve. Methods marked with +abst use features that are not present in low resource languages.

	GDC	Schiller	TSV	poems	MHG
base	0.26	0.32	0.70	0.74	0.22
iter 1	0.60	0.44	0.84	0.77	0.61
iter 2	0.71	0.53	0.67	0.74	0.25
iter 3	0.46	0.55	0.72	0.78	0.60
iter 4	0.73	0.62	0.70	0.77	0.40
iter 5	0.95	0.70	0.59	0.78	0.60
iter 6	0.60	0.77	0.70	0.82	0.66

Table 2: Results of the iteratively trained model on the GerDraCor (GDC) and Schiller test sets (precision at top 100) and on the TSV and poetry test sets (average precision); The MHG column shows the results on the Middle High German test set (precision at top 100).

the SVM we set the regularization term C to 1.0 and $gamma$ to $auto$. We normalized the features by substracting the mean and dividing by their variance. The baseline features contain an abstractness feature which may not be present in low resource languages. To enable a fair comparison, we used these features both with and without the abstractness feature present and trained SVMs for each approach. Table 1 shows that our supervised approach achieves similar results to the supervised baseline features together with the abstractness. Without abstractness, our approach achieves a higher average precision by 0.13 percent points on the TSV set, while staying in a similar range on the poems set. The baseline results without the abstractness feature on the poems set is interesting, since it even surpasses the baseline with all features present. Our results show that our approach can utilize the information contained in the word embeddings more efficient than the baseline, while we do not need to use the abstractness feature.

4.4. Unsupervised metaphor retrieval

In this experiment, we again used the annotated TSV metaphor dataset and the poems dataset. However, we did not use any examples annotated as metaphors for our zero-shot approach. As explained in Section 3, we used randomly connected adjectives and nouns from the GerDraCor training set as metaphor training examples in one approach. In another approach we used random combinations of the TSV and poems training sets as training. Results in Table 1 (marked as *zero-shot*) show that we get slightly lower average precision than the baseline approach with the abstractness features when using unsupervised GerDraCor pretraining. However, we get far better avarage precision numbers than the baseline approach without the abstractness features when using this pretraining. When the abstractness features are used – which are not available in low resource languages – our approach reaches a lower or similar average precision to the baseline. This shows that our method is especially useful in a low resource language context when no additional features are present, while still remaining in a similar range for languages with more resources.

4.5. Case studies

Our main goal is a method to generate a metaphor dataset and create a metaphor retrieval system for a low resource language with no previously annotated metaphor dataset. To analyze whether our approach is suitable for this, we conducted two case studies: One on German and one on Middle High German.

Setup For the German texts we extracted adjective-noun pairs from one half of the GerDraCor corpus and used them to train the unsupervised zero-shot system. Two sets of random combinations of adjectives and nouns were used as pseudo metaphor examples. Additionally we separated the 11 texts by Friedrich Schiller contained in the GerDraCor corpus to analyze the metaphor detection rates on the works of a single author. For the MHG data we used eleven texts from the Mittelhochdeutsche Begriffsdatenbank to extract word pairs. In every iteration we then annotated the top 100 rated unannotated examples in the training corpus, the bottom 50 unannotated examples and another random 50 unannotated examples. This strategy allows to build a metaphor training dataset for both of these languages while finetuning the classifier on the new data. We discarded multiple occurrence of the same word pairs as well as ambiguous examples and detections based on errors like wrong PoS tagging. For German, the final training dataset contained 390 metaphors and 449 non-metaphors, for MHG it was 287 metaphors and 365 non-metaphors, respectively. To test our approach, we used our trained models to rank the candidates in the remaining corpora by their metaphoricity. We annotated the top 100 results on the other half of the GerDraCor corpus for German and the top 100 results on eleven other texts from the Mittelhochdeutsche Begriffsdatenbank for MHG. Additionally we tested our approach for German on an extra held out dataset from GerDraCor, comprising only the works by Friedrich Schiller, to evaluate our model on a single author from a more recent period.

Results The results in Table 2 show that the zero-shot classifier found 26 metaphors in the general top 100 results for German, 32 metaphors for the works of

Schiller, and 22 metaphors in the top 100 results for MHG. After only one round of annotation, this already increased to 60 metaphors for German, 44 for Schiller and 61 metaphors for MHG. This shows that even with minimal annotation effort, the unsupervised pretraining together with our candidate mining strategy provide a useful model for metaphor detection. However, it can also be seen that for the heterogenous corpora and further iterations this process is still not completely stable. While a tendency towards improvement can be seen, further investigations are necessary. For the single author study on the works of Friedrich Schiller, we see that the results improve with every iteration of finetuning, reaching 77% from an inital 32%.
Below you can find examples of found metaphors in German (DE) and Middle High German (MHG):

grenzenloses Mitleid **borderless sympathy**	(DE)
ein **aufrichtiges Herz** an **upright heart**	(DE)
Behutsam schreite her auf **leisen Sohlen** Gentle shall he tread on **silent soles**	(DE)
schoenen gewin **radiant victory**	(MHG)
der vogele **süezer dôz** the birds' **sweet sound**	(MHG)
mit vil **getriuwer huote** with much **faithful loyalty**	(MHG)

5. Limitations

While our approach uses only minimal additional information, POS tags are still needed to find the metaphor candidates. The approach also relies on word embeddings, which have to be trained on the available low resource data. Since the available corpora may not always represent the use of language completely, especially for low resource languages, there is always the danger that the word embeddings do not correctly encode the semantic information of the words, e.g. due to common words in a language occuring only infrequently in the corpus used for training. This may be mitigated to some point by our model, which transforms the word vectors into another space, instead of directly using the word embeddings.

6. Conclusion

In this work, we presented a novel unsupervised method to enable metaphor detection. We demonstrated that our approach improves over comparable baseline approaches. The design of our method allows us to apply it to low resource languages without changes. It produces excellent results when used in a supervised manner. While the results are worse when the method is used without labeled data, the method can still be used to enable a bootstrapping approach. Metaphor candidates are extracted from a text in an unsupervised manner, labeled, and then used to train the supervised method. Thus, our approach on the one hand enables metaphor detection in uninvestigated low resource languages, and on the other hand serves as a powerful supervised tool once the first metaphors have been discovered. An interesting next step would be to combine our approach with other unsupervised approaches mentioned in the related work section that are applicable for low resource languages.

7. Bibliographical References

8. Language Resource References

Fischer, Frank and Börner, Ingo and Göbel, Mathias and Hechtl, Angelika and Kittel, Christopher and Milling, Carsten and Trilcke, Peer. (2019). *Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama*. Zenodo.

Klein, Thomas and Wegera, Klaus-Peter and Dipper, Stefanie and Wich-Reif, Claudia. (2016). *Referenzkorpus Mittelhochdeutsch (1050-1350), Version 1.0)*.

Zeppezauer-Wachauer, K. (2022). Mittelhochdeutsche begriffsdatenbank (mhdbdb). Universität Salzburg. Interdisziplinäres Zentrum für Mittelalter und Frühneuzeit (IZMF). Koordination: Katharina Zeppezauer-Wachauer. 1972-2022 (laufend). URL: http://www.mhdbdb.sbg.ac.at/ (12.04.2022).