“Japan’s Answer to Mozart”: Automatic Detection of Generalized Patterns of Vossian Antonomasia

Michel Schwab[1], **Robert Jäschke**[1,2] **and Frank Fischer**[3]
[1]Humboldt-Universität zu Berlin, Germany
[2]L3S Research Center, Hannover, Germany
[3]Freie Universität Berlin, Germany
{michel.schwab,robert.jaeschke}@hu-berlin.de
fr.fischer@fu-berlin.de

Abstract

Vossian Antonomasia (VA) is a rhetorical device used to describe an entity (the target) by transferring certain features and characteristics of another entity (the source) to it. The phenomenon is closely related to metaphor and metonymy. Similar to these more familiar devices, the detection of VA expressions is a challenging task. We propose novel VA detection models that center on the source to tackle this problem. The focus lies on the ability of the models to detect VA independent of the syntactic patterns they appear in. We model the problem in different scenarios and utilize a state-of-the-art metonymy resolution model that relies on word masking, and metaphor detection models, which are based on linguistic metaphor theories, and adjust them to our task. All models leverage pre-trained language models such as BERT and RoBERTa. As there is limited annotated data available, we use a data augmentation technique to create a new dataset consisting of VA with new syntactic patterns where the generalization ability of the models can be evaluated.

1 Introduction

Vossian Antonomasia (VA) is a stylistic device that refers to an entity by naming another famous named entity that shares certain characteristics or sets of attributes with the entity. In general, it consists of three chunks (Bergien, 2013): The *target* is the entity which is being described. The *source* is the famous entity that typically stands for a certain set of attributes. The *modifier* is the component that shifts the characteristics of the source to the target’s environment. When Angela Merkel is referred to as “the German Margaret Thatcher” (Trippe, 2005), “Angela Merkel” is the target entity that inherits one or more attributes from the source entity, in this case from the Iron Lady, “Margaret Thatcher”. The modifier (“German”) projects these attributes traditionally associated with Margaret Thatcher onto Angela Merkel. The combination of source and modifier is called a *VA phrase* in the following.

To understand VA, one requires a deep cultural and historical knowledge of the source entity, as the transferred characteristics are often not explicitly mentioned, but only indicated by the name of the source that stands for the attributes. Thus, the readers themselves must infer the author’s intention. This can be achieved by the context the expression appears in and knowledge about the source itself. The context is quite important because, in most cases, an entity does not only stand for one property. Arnold Schwarzenegger serves as a good example of a person who successfully moved between fields and changed the characteristics and attributes he stands for. First, he was known as a successful bodybuilder, but after turning to acting and politics, the focus of his persona shifted to his newly achieved accomplishments and his ability to successfully transition between fields.

The automatic detection of VA is challenging as their syntax is often ambiguous and hard to distinguish from literal expressions. See, for example, “the German Angela Merkel” vs. “the American Angela Merkel” (Pohl, 2016). The first phrase is literal stating that Angela Merkel is a person from Germany. In contrast, in the second phrase, Angela Merkel stands for a set of characteristics and is used as a source in a VA expression to describe Hillary Clinton.

Recent years have seen various approaches to the automatic detection and extraction of VA from larger text corpora. The first steps were pattern-based approaches (Jäschke et al., 2017; Fischer and Jäschke, 2019; Schwab et al., 2019), but recently language models like BERT (Devlin et al., 2019) were employed (Schwab et al., 2022). They achieved strong results and are also robust on unseen data.

In this paper, we tackle the problem of detecting VA expressions independent of their syntactic structure. The syntactic structure of VA phrases consisting of source and modifier can include a wide range of variations. So far, the only annotated VA dataset (Schwab et al., 2023b) consists solely of examples where the modifier follows the source, i.e., “the SOURCE of MODIFIER”, which is a commonly used syntactic pattern for VA phrases. This is because of the variety of naming the modifier. In comparison to other syntactic patterns, the modifier in this pattern can have an arbitrary length and complex structure. In contrast, other patterns such as “the MODIFIER SOURCE” (see Table 1 for more syntactic patterns) often impose stricter limitations, typically requiring the modifier to be a single word, such as an adjective or noun. To our knowledge, there is no study describing the extraction of VA where the modifier precedes the source. We address the problem of a generalized VA detection approach by focusing solely on the source during training to remove the boundaries associated with the modifier and target. To achieve this, we develop five different methods. One is a sentence classification model that uses special tokens to indicate the candidates. The next is based on a sentence-pair classification model. Two rely on linguistic metaphor theories. We adapt the metaphor theories to the source of VA expressions, since source entities in text, like metaphorical words, are not meant literally. The last method is an adaptation of a metonymy resolution model, since VA is often categorized as a specific subtype of metonymy.

Next to getting a deeper understanding of the phenomenon itself, the detection of VA can support various NLP tasks. It can provide new and interesting question answering challenges, as Schwab et al. (2023a) have shown that one can easily transform the combination of source and modifier (VA phrase) into questions. Schwab et al. (2023a) also showed that VA phrases are hard to be captured correctly by coreference resolution models. The models must understand that the source is not independent, but a part of the target’s reference chain. Often, this did not work and the sources were predicted in new standalone reference chains. Thus, VA detection could improve and support coreference resolution. By understanding figures of speech like VA, language models can better understand natural language in general and especially the nuances of human language. With that, more human-like text could also be generated, for instance, spiced-up headlines for newspaper articles.

This paper is structured as follows: In Section 2, we discuss related work, while in Section 3, we present the datasets and explain the dataset generation process in detail. In Section 4, we describe the developed models and methodology, followed by an empirical evaluation of the proposed models in Section 5. Finally, Section 6 closes the paper with a conclusion.

Our code and data are freely available.[1]

2 Related Work

The research on the automatic detection of VA is a relatively new topic in the NLP area. There exist multiple approaches on the (semi-)automatic detection and extraction of the phenomenon that have been developed recently. Jäschke et al. (2017), Fischer and Jäschke (2019) and Schwab et al. (2019) used semi-automatic approaches that were based on syntactic patterns around the source. In particular, they used regular expressions to extract candidate sentences from a newspaper corpus and matched those candidates against entity lists. Fischer and Jäschke (2019) and Schwab et al. (2019) removed common false positives in a second step using a manually curated blacklist. While Schwab et al. (2019) additionally presented a first fully automatic approach for VA detection employing a bidirectional long short-term memory (BLSTM) network, in Schwab et al. (2022) the approaches using neural networks were more advanced. They used concatenations of BLSTM and attention layers with ElMo embeddings (Peters et al., 2018) as well as a fine-tuned BERT model (Devlin et al., 2019) for binary sentence classification. Additionally, they presented a VA tagger that tags all parts of a VA expression in a sentence employing BLSTM and conditional random fields as well as a fine-tuned BERT model for sequence tagging. Schwab et al. (2023a) did not detect VA expressions, but tackled the task of detecting the target entity inside the newspaper article in which the VA expression appeared, which was neglected in the previous approaches. They showed that by transforming a VA phrase into a question, a hybrid model that sequentially uses a QA model and a coreference resolution model could yield high scores without fine-tuning the models further.

Similar tasks like metaphor detection have been studied deeply. Most of the research is based on

[1] `https://vossanto.weltliteratur.net/icnlsp2023/`

the Mozart of Japan
Japan’s (the Japanese) Mozart
Japan’s (the Japanese) answer to Mozart
Japan’s (the Japanese) version of Mozart
Japan’s (the Japanese) equivalent of Mozart

Table 1: An example of the data augmentation versions with nouns (Japan) and their adjective forms as modifiers in brackets (Japanese). In total, we get eight additional versions per sentence.

neural networks. While Gao et al. (2018), Dankers et al. (2019) and Torres Rivera et al. (2020) used sequence tagging models based on contextual word embeddings, other models are focusing on single word classification. In particular, they classify words according to whether they are meant literally or metaphorically. Choi et al. (2021) make use of two linguistic metaphor theories which they implement by employing a pre-trained language model, RoBERTa, and extract the embeddings in context and without context to train a multilayer perceptron (MLP). Most recently, Wang et al. (2023) follows the idea of Choi et al. (2021), but additionally focuses on selecting relevant context for the classification task employing a dependency parser for denoising the context around the candidate word which works especially well on long input sentences.

Metonymy resolution is another similar task that has been researched, especially recently with the use of pre-trained language models (Su et al., 2020; Li et al., 2020; Mathews and Strube, 2021). Often, the task is limited to location metonymy resolution (Li et al., 2020; Su et al., 2020). While Li et al. (2020) models the task as a token-level classification task, Mathews and Strube (2021) introduces a sequence tagging approach. Both models mask their candidates during training and evaluation.

3 Data

Candidate Generation We use the dataset from Schwab et al. (2023b) for training. There, we need to identify phrases that are candidates for VA sources. As, by definition, the source of any VA expression has to be a named entity, we utilize a state-of-the-art named entity tagger, FLAIR (Akbik et al., 2019), to obtain all candidate entities for each sentence in the dataset. We then collect tuples for each sentence consisting of an entity in the sentence and the sentence itself. The tuples containing a source entity are labeled positive, all others negative.

We remove candidates where the text sequence the NER tagger has identified as entity mention does not exactly match a source phrase, as those cases are difficult to handle correctly (and only a small part of the data is affected, cf. Sec. 5). Consider the sentence “He is the Michael Jordan of swimming, but he was never as good as Michel Jordan.”. If the tagger would only tag “Michael” or “Michael Jordan of swimming” as an entity we remove those candidates.

The sentence highlights another issue: an entity that is mentioned more than once in the same sentence can not be distinguished within our set of tuples. When all mentions are no VA sources, we keep the tuples. When one of those entity mentions is indeed a source, we keep that tuple (i.e., the tuple with the positive label) and remove the others, since such cases are very rare. We removed 38 negative tuples in the training data and nine negative tuples in the test data.

Training Data Compared to other more popular rhetorical devices, such as metaphor and metonymy, one challenge of VA detection is the lack of annotated data. The only annotated English VA dataset is, to our knowledge, the one by Schwab et al. (2023b). It was first introduced by Schwab et al. (2019) and later annotated further (Schwab et al., 2022, 2023a). The dataset contains sentences from the New York Times Annotated Corpus (Sandhaus, 2008). The corpus contains articles from the New York Times from 1987 to 2007, comprising around 60,000,000 sentences. The dataset was created in a semi-automated way. First, frequently used syntactical patterns around the source were identified and candidate sentences extracted. The patterns consist of one of the words before (`the/a/an`) and after the source (`of/for/among`). Using all possible combinations, the authors of Schwab et al. (2019) obtained nine different patterns. Then the words between these combinations were matched against an entity list, and finally those candidates were checked against a manually curated black list to remove false positives and manually labeled. In total, the dataset contains 6,095 sentences of which 3,115 include VA expressions. On this dataset we generate candidates as explained before, which results in a training dataset of 16,877 sentence-entity tuples with 2,868 positive instances (17%) and 14,009 negative instances.

Test Data The lack of syntactic variations of VA expressions is one significant issue for testing VA detection models on generalization. For example, in the training data the modifier always appears directly after the source:

```
(a|an|the) SOURCE (of|for|among) MODIFIER.
```

This pattern, however, does not cover all variants of VA as the syntactic patterns in which source and modifier appear are more diverse. Annotating a text corpus to identify new syntactic variations is prohibitively expensive due to the rarity of the phenomenon on the sentence level (Schwab et al., 2019). Another approach is syntactic data augmentation which changes the syntax of a sentence without affecting its semantics. In our case, it is especially crucial to ensure that the VA expressions remain intact and that their meaning is not changed.

To augment data, we identified eight different VA patterns consisting of source and modifier that are different from the ones in the training data. In particular, their source follows *after* the modifier. Two of the patterns have no words between the modifier and source (named "—" in the sequel), the other six patterns have connecting words between modifier and source. We refer to these phrases ("answer to","version of", "equivalent of") as "connector phrase" (CP). Each of the phrases appears two times in the patterns.

The first four patterns are represented by the following regular expression and involve a modifier that is a noun:

```
MODIFIER's (CP)? SOURCE,
```

where CP is a connector phrase, MODIFIER is the modifier chunk and SOURCE is the source entity. The remaining four patterns include a modifier that is an adjective and are represented by the following regular expression:

```
(a|an|the) MODIFIER (CP)? SOURCE,
```

where the choice of the article at the beginning ("a", "an" or "the") depends on the article in the original VA phrase.

Six of the eight patterns include a CP between modifier and source consisting of two words, whereas the other two patterns do not have any words between them. This is another distinction from the pole word succeeding the source in the annotated data. Furthermore, the grammatical category of the modifier changes. While 92% of the modifiers in the annotated data are noun phrases,[2] this is not the case for the last four patterns, where the modifier is an adjective.

The modifiers in the annotated data can be complex (the longest modifier consists of 25 words) and not all VA phrases can be easily augmented as they cannot be changed semantically correct into a noun or adjective. Thus, we use a subset of the data for augmentation. Specifically, we extract all sentences that include a VA expression where the modifier is a geographical place that possesses an adjectival form. This ensures that the meaning is not changed when adapting the modifier and the syntax. We achieve this using the lists of adjectival and demonymic forms of place names from Wikipedia.[3] This has the advantage that those modifiers can always be transformed into any of the eight patterns since place names are always nouns and have an adjectival form which is suitable for both, noun and adjective modifiers.

Hence, we match all modifiers against the lists. In total, we could extract 244 VA expressions where the modifier matches an entry in one of the Wikipedia lists. Countries were mentioned most often (159), followed by cities (51), regions (23), and continents (11).

By augmenting each sentence, we obtain 1,952 augmented sentences (244 per pattern), which we call *augmented data*. Along with the 244 original instances (*original data*), this yields a total of 2,196 sentences. Again, we apply the candidate generation method and compute entity-sentence tuples for each sentence. In total, we produced 8,480 unique instances of which 2,196 are positive and 6,284 are negative. The positive label ratio increases compared to the positive label ratio of the training data to 26% as each sentence in the test data consists of a VA source which is not the case in the training data. Each sentence produced on average 34.5 instances with a standard deviation of 16.1 which shows that the number of generated instances per sentence is quite diverse depending on the number of named entities.

4 Methods

As explained before, the anchor of a Vossian Antonomasia expression is the source entity which is being invoked as a point of comparison. Thus, we

[2] Which we detected with the dependency parser from spaCy (Honnibal et al., 2020).

[3] https://en.wikipedia.org/wiki/List_of_adjectival_and_demonymic_forms_of_place_names

aim to find source entities of VA expressions using different approaches. In particular, our goal is to identify generalized VA expressions across diverse syntactic structures. As a baseline, we adapt a state-of-the-art VA extraction model. We then make use of models that were successfully applied in similar areas. In particular, we adjust two metaphor detection models which are based on linguistic metaphor theories. Similar to metaphorical words, the source entity of a VA expression is not meant literally, but stands for a set of characteristics. Additionally, we adapt a metonymy resolution model that uses candidate word masking. Finally, we present two fine-tuned RoBERTa models for sentence(-pair) classification which are designed to focus on the entity candidates inside a sentence. Contrary to the baseline, which is a sequence tagging model, all subsequent models are binary classification models. The goal is to determine whether pre-computed entities serve as a source of a VA expression or not. As explained in Section 3, we first identify all entities in a sentence and then classify each entity-sentence tuple.

BERT_SEQ This baseline is an adaptation of the sequence tagging model in Schwab et al. (2022), BERT_SEQ. Each word in a sentence is tagged to determine whether it is part of a chunk of a VA expression (i.e., target, source, or modifier) or not. It is a fine-tuned BERT (base-cased) model which outperformed a BLSTM-CRF model.

Here, we modify this model by focusing on tagging the source words only, employing the IOB tagging scheme. In particular, like Schwab et al. (2022), we add an additional linear layer to the BERT model to tag all words in a sentence. This enables a better comparability with our newly developed models. Additionally, the implicit focus on the order of chunks will vanish, which potentially leads to better generalization.

BERT_MASK This model is an adaptation of a state-of-the-art location metonymy resolution model (Li et al., 2020). The model is based on the idea that the context is more important to distinguish between metonymic and literal usage than the potential metonymic candidate itself. We use this model, since VA is often categorized as a subtype of metonymy, and apply it to our source candidates. In particular, we follow Li et al. (2020) in that we mask the source candidates with the single token *X* during training and evaluation. Further, we fine-tune a BERT model for word-level classification. Specifically, we extract the embeddings of the masked token (*X*) from the last hidden layer of BERT and feed them into a binary linear classifier to classify whether the candidate is a VA source. In cases where the masked token is omitted in the preprocessing due to truncation, we utilize the *[CLS]* token for classification instead.

RoBERTa_MIP This model is inspired by the Metaphor Identification Procedure (MIP) (Group, 2007). In short, the theory proposes that a word is used metaphorically when its literal meaning deviates from its contextual meaning. The key is that the literal meaning of the word would not apply directly in the used context. We transfer this concept to the source entity of a VA expression: The source entity is not meant literally, but is a placeholder for a set of characteristics the entity stands for. The entity would normally not be used in the context, as the target and especially the modifier are normally not directly related to the source. Thus, we state that a named entity used as a source in a VA expression has a different meaning in the context (a set of characteristics) from its more basic meaning (the entity itself).

We then roughly follow Choi et al. (2021). As base, we utilize RoBERTa (Liu et al., 2019), a pretrained language model. First, each word of a sentence is tokenized. The special character sequences `<s>` and `</s>` are then added at the beginning and end of the token sequence, respectively. Next, we compute the position embedding which represents the position of each token within the sentence and the segment embedding which indicates which tokens belong to the candidate. The input embeddings are finally obtained by the element-wise addition of token, position embedding, and segment embedding. In a separate step, we tokenize the isolated candidate words using the same tokenizer and also add special characters accordingly.

The embeddings of the candidate tokens in both steps are averaged independently. This results in two embeddings: the contextualized embedding and the isolated embedding for the candidate. These embeddings are then concatenated and passed through a linear layer that outputs a binary label indicating whether the entity is a VA source or not.

RoBERTa_SPV The model is based on Selectional Preference Violation (SPV) (Wilks, 1975, 1978), which is popular in metaphor detection methods, see Mao et al. (2019) and Choi et al. (2021). The idea of SPV in the context of metaphors is that a word is metaphorical when it appears unusual in its surrounding context, that is, it typically does not co-occur with the surrounding words. We adapt this theory to VA detection: An entity serving as a source for VA expressions appears unusual within its surrounding words, especially with the modifier which normally represents an environment unrelated to the source. Instead, the modifier is connected to the target entity.

As in the MIP model, we compute tokens, position embedding and segment embedding of the sentence. Subsequently, we compute the contextualized embedding accordingly and also calculate the embedding of the special <s> token, which represents the aggregated representation of the sentence. Both embeddings are concatenated and passed through a linear layer which returns a binary label.

RoBERTa_CLF We adapt the binary sentence classification model from Schwab et al. (2022). Specifically, we introduce two special tokens, `[START_SRC]` and `[END_SRC]`, to denote the start and end, respectively, of the candidate by encasing the source entity inside the sentence with both tokens. These tokens are added to the tokenizer. The adapted sentence is then used as input RoBERTa which we fine-tune for binary sentence classification by adding and training a linear layer.

RoBERTa_PAIR For this model, we reformulate the task as a sentence-pair classification problem. This task is typically used to assess the relationship between two sentences, such as next sentence prediction, contradiction of sentence-pairs or semantic relations. In our case, the first sentence consists of the candidate entity only, while the second sentence provides context in form of the corresponding sentence the candidate entity appears in. We want the model to learn to classify whether the candidate entity is a source in the corresponding sentence or not. As in RoBERTa_CLF, we adapt RoBERTa by appending a linear layer on top of the RoBERTa model. Subsequently, we fine-tune the model for binary classification.

5 Evaluation

In this section, we describe the experimental settings before presenting and analyzing the empirical results of our models. All models rely on the output of an NER tagger whose output is used to form the set of source candidates. This is different from the baseline model, which does not need candidates but classifies each word individually.

The tagger we used in our study has an F_1 of 0.94 on the CoNLL-03 dataset. In our specific case, it missed identifying 110 (3.8%) out of 2,868 source entities in the training dataset and 40 (1.8%) out of 2,196 source entities in the test dataset. For the sake of comparability, we exclude these instances from the evaluation process. The idea behind the exclusion is that we aim to evaluate the individual performance of our models rather than the whole performance of the NER tagger combined with our models. We use precision, recall and F_1 score to assess the performance of the models.

5.1 Experimental Settings

We conduct hyperparameter optimization on dropout rate, epochs, learning rate, and batch size based on F_1 score.[4] For this, we use 25% of the test data as a validation set for all models including the baseline. We use a part of the test data for hyperparameter optimization on purpose as we want to determine the best model for the generalized test data. We assume that if the model works well on the test data, it should still be able to achieve good performance on the data we trained it with. We will evaluate this in the subsequent section.

We use the pre-trained BERT base-cased model[5] as in Schwab et al. (2022) for BERT_SEQ as well as for BERT_MASK, and the pre-trained RoBERTa base model[6] for all other models as basis. Both models share the same architectural parameters. Specifically, each model has 12 transformer blocks, 12 attention heads, and the dimensionality of the hidden states is set to 768. For all models, we use AdamW optimizer (Loshchilov and Hutter, 2017). We implemented our models using the Hugging Face framework and PyTorch. The code is free available on our website.[7]

[4] See Appendix A for details and final choices for each model.

[5] `https://huggingface.co/bert-base-cased`

[6] `https://huggingface.co/roberta-base`

[7] `https://vossanto.weltliteratur.net/icnlsp2023/`

model	precision	recall	F_1
BERT_SEQ	.88 ±.02	**.97** ±.03	**.92** ±.02
BERT_MASK	.83 ±.03	.88 ±.02	.85 ±.02
RoBERTa_MIP	.88 ±.02	.89 ±.04	.88 ±.01
RoBERTa_SPV	**.93** ±.03	.85 ±.07	.89 ±.03
RoBERTa_CLF	.87 ±.03	.87 ±.02	.87 ±.02
RoBERTa_PAIR	.76 ±.04	.94 ±.02	.84 ±.01

Table 2: Performance of the models using 5-fold cross validation on the training dataset.

5.2 Results on Training Data

Table 2 presents the results on the training data using stratified 5-fold cross validation. All approaches, even if hyperparameters were not optimized for this data, achieve strong results. Surprisingly, the baseline, BERT_SEQ, has the best results, having an F_1 score of 0.92, although the gap to the other models is not large. RoBERTa_CLF and both adapted metaphor detection model, RoBERTa_MIP and RoBERTa_SPV, have similar scores of 0.87, 0.88, and 0.89, respectively. Only BERT_MASK and RoBERTa_PAIR achieve a little lower score of 0.85 and 0.84, respectively. The general high scores are expected as the models were trained on similar data regarding the syntax of the VA expressions. Also, the label ratio is the same as we conducted stratified sampling for the cross validation.

5.3 Zero-shot Results on Test Data

We conduct a zero-shot transfer with our models on the test data consisting of the original and augmented data as explained in Section 3. This evaluation is conducted to analyze how the models generalize to new syntactic VA variations which is the main goal of our work. In this evaluation, we obtain surprising results. While the performance of all models except RoBERTa_PAIR decreases drastically in all metrics, RoBERTa_PAIR increases its performance to an F_1 of .86 (cf. Table 3). While the precision of RoBERTa_PAIR increases substantially, the recall decreases. The other models are not able to compete against this model. Even the results of the second best model, RoBERTa_MIP, decreases to an F_1 score of 0.74 which is a gap of 0.12 points. Still, this is the smallest performance gap and shows that the adaptation of the MIP theory works better for generalized VA detection than the rest of the models. BERT_MASK attains the lowest F_1 score of 0.25. The baseline, which achieved the best results on the training data, is also not able to solve this task with the second lowest F_1 of 0.27 as well as RoBERTa_SPV and RoBERTa_CLF whose scores also dropped to 0.58 and 0.41, respectively.

The performance on the original data in the test dataset (713 instances, 183 positive) even increases for all models compared to the results on the training data. This is not that surprising, as the syntax is the same as in the training data. Also, the training data consists of sentences without VA expressions that are syntactically very similar to those with VA expressions. In the test data, however, there exist no such negative examples. Thus, this might be a reason why the scores are rising. The performance on the augmented data drops dramatically in almost all models compared to the performance on the training data (cf. Section 5.2).

This shows that only RoBERTa_PAIR is able to handle new syntactic variations in contrast to all other models. As the F_1 almost did not change between the evaluation on both datasets, it shows a robustness to new data. The metaphor detection models had similar scores on the training data and could obtain high scores on the original data, but they diverge on the augmented data. While RoBERTa_SPV drops to an F_1 of 0.53, which is 0.36 points less than on the training data, the performance gap of RoBERTa_MIP is smaller.

In general, it seems that in all models except RoBERTa_PAIR, the syntax of the VA expression still has a major influence on the correct classification.

Performance vs. Syntax (RoBERTa_PAIR) An interesting point to investigate further is the influence of the syntactic variations we used for data augmentation. In total, we have four syntactic patterns, three that consist of the connector phrases between modifier and source, “answer to”, “version of”, “equivalent of”, and the “—” version without any connector phrase between both chunks. Table 4 shows the results in the ‘total’ block. We can see that all three metrics are best for the pattern “—”, with an F_1 of 0.92. For the “equivalent of” and “version of” patterns, the model still achieves high scores, whereas for the “answer to” patterns, it performs worse with an F_1 of 0.72 which is 0.2 lower than the best score. It is interesting that one pattern is much harder to detect and shows that even if patterns seem quite similar for humans, it is much

	total			original data			augmented data		
model	precision	recall	F_1	precision	recall	F_1	precision	recall	F_1
BERT_SEQ	.73	.17	.27	**.99**	**1.00**	**.99**	.42	.07	.12
BERT_MASK	.69	.15	.25	.92	.81	.86	.52	.07	.13
RoBERTa_MIP	.92	.62	.74	.97	.92	.95	.91	.58	.71
RoBERTa_SPV	**.97**	.42	.58	**.99**	.87	.93	**.97**	.36	.53
RoBERTa_CLF	.73	.29	.41	.95	.82	.88	.66	.22	.33
RoBERTa_PAIR	.89	**.83**	**.86**	.91	.97	.94	.89	**.81**	**.85**

Table 3: Performance of the models on the test data which include the original and augmented data.

	total			modifier is a noun			modifier is an adjective		
syntax	precision	recall	F_1	precision	recall	F_1	precision	recall	F_1
—	**.91**	**.93**	**.92**	**.90**	**.93**	**.92**	**.91**	**.93**	**.92**
answer to	.85	.62	.72	.88	.72	.79	.82	.51	.63
version of	.89	.83	.86	**.90**	.90	.90	.87	.77	.81
equivalent of	.89	.87	.88	**.90**	.91	.91	.87	.82	.85
total	.89	.81	.85	.90	.87	.88	.87	.76	.81

Table 4: Performance of RoBERTa_PAIR on the augmented data, split up by pattern and POS type.

harder for the models to detect them correctly.

Performance vs. POS (RoBERTa_PAIR) We now analyze whether the part of speech (POS) tag of the modifier influences the model using the best performing model on the test dataset, RoBERTa_PAIR. One half of the augmented data has modifiers that are adjectives whereas the other half has modifiers that are nouns. In Table 4, we can clearly see that the model performs better when the modifier is a noun with an F_1 of 0.88 compared to 0.81 for the adjective examples. One reason is the high performance gap of 0.11 in recall. A plausible reason is the fact that in the training data, the modifiers of the VA expressions are almost always noun phrases (and thus include at least a noun), which possibly is captured in the fine-tuning process, even if the modifier is not marked explicitly.

5.4 Error Analysis (RoBERTa_PAIR)

In total, we got 851 false positive and 159 false negative errors in the 5-fold cross validation. In 239 cases, an entity candidate was falsely predicted as source entity in a sentence that included a VA expression. Still, in the majority (612) of the false positive errors, the entities appeared in a sentence without any VA occurrence.

In the test dataset, more false negative errors (281) than false positives (172) occurred. Grouping the false positives by entity and original sentence, we got 25 groups where in 14 of them all augmentations with the same candidate were predicted falsely. The false negatives, on the other hand, grouped into 80 groups, which makes sense as the syntax around the source entities changed, whereas the syntax around the entity candidates in the false positive instances did not and thus, the model's prediction should be more similar.

Table 5 shows a sample of false positive and false negative errors, the RoBERTa_PAIR model did in the test dataset.

The false positives included candidate entities that belonged to the VA expression but as a target chunk ("Manno Charlemagne") or as a modifier chunk ("European"). That was expected as they are somehow connected semantically to the source and thus, it is harder for the model to differentiate between them. It also appeared that an entity that was used as source ("Berlusconi") was also mentioned in another typing elsewhere in the sentence with a literal meaning ("Silvio Berlusconi"). Those examples are rare as the source is normally not mentioned in the context. Still, these are decisions that are especially hard to predict correctly for the sentence pair model as the model has no explicit focus on the position of the entity in the sentence as the other models had.

Ex. 1: He doesn't want to be Syria's version of **Gorbachev**.

Ex. 2: "He's the Japanese answer to **Cal Ripken**, but with more punch," said Marty Kuehnert, a sports broadcaster and longtime resident in Japan.

Ex. 3: Buena Vista Home Entertainment, the distribution arm of Disney, recently acquired a library of Japanimation created by a man often hailed as "the **Walt Disney** of Japan," Hiyao Miyazaki.

Ex. 4: One of the anthology's strongest cuts, "Ayiti Pa Fore" ("Haiti Is Not a Forest') was recorded in 1988 and features **Manno Charlemagne**, a singer and songwriter who is regarded as Haiti's answer to Bob Marley.

Ex. 5: In the capital, intellectuals refer to Mr. Thaksin as Asia's Berlusconi, a reference to Prime Minister **Silvio Berlusconi** of Italy, a business tycoon who has faced continuing accusations of conflict of interest.

Ex. 6: Its chairman, Jan Carlzon, is credited with turning the airline around in the early 1980s, earning a reputation as "the **European** answer to Lee Iacocca," one analyst said.

Table 5: Incorrectly classified instances of RoBERTa_PAIR on the test dataset. False negatives (Ex. 1-3) are marked green, false positives (Ex. 4-6) red.

6 Conclusion

We proposed four novel VA detection models and analyzed their ability to detect generalized VA expressions across a range of syntactic patterns. To achieve this, we use data augmentation techniques to create a VA dataset including numerous new syntactic patterns. We develop VA detection models based on adjusted linguistic metaphor theories and a metonymy resolution model that are applied to the source. While most models struggle to generalize well to these new patterns, our best model, RoBERTA_PAIR, achieves good results on both, the training and test dataset.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Angelika Bergien. 2013. Names as frames in current-day media discourse. In *Name and Naming. Proceedings of the second international conference on onomastics*, pages 19–27, Cluj-Napoca. Editura Mega.

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

Verna Dankers, Marek Rei, Martha Lewis, and Ekaterina Shutova. 2019. Modelling the interplay of metaphor and emotion through multitask learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2218–2229, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Frank Fischer and Robert Jäschke. 2019. 'The Michael Jordan of greatness'—Extracting Vossian antonomasia from two decades of The New York Times, 1987–2007. *Digital Scholarship in the Humanities*, 35(1):34–42.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 607–613, Brussels, Belgium. Association for Computational Linguistics.

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Robert Jäschke, Jannik Strötgen, Elena Krotova, and Frank Fischer. 2017. “Der Helmut Kohl unter den Brotaufstrichen”. Zur Extraktion Vossianischer Antonomasien aus großen Zeitungskorpora. In *Proceedings of the DHd 2017*, DHd ’17, pages 120–124. Digital Humanities im deutschsprachigen Raum.

Haonan Li, Maria Vasardani, Martin Tomko, and Timothy Baldwin. 2020. Target word masking for location metonymy resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3696–3707, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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, Florence, Italy. Association for Computational Linguistics.

Kevin Alex Mathews and Michael Strube. 2021. Impact of target word and context on end-to-end metonymy detection.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ines Pohl. 2016. "America’s fear has a new name: merkel". Accessed on 06 23, 2023.

Evan Sandhaus. 2008. The New York Times Annotated Corpus LDC2008T19. DVD, Linguistic Data Consortium, Philadelphia.

Michel Schwab, Robert Jäschke, and Frank Fischer. 2022. "The Rodney Dangerfield of Stylistic Devices": End-to-end detection and extraction of vossian antonomasia using neural networks. *Frontiers in artificial intelligence*, 5.

Michel Schwab, Robert Jäschke, and Frank Fischer. 2023a. “who is the madonna of Italian-American literature?”: Target entity extraction and analysis of vossian antonomasia. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 110–115, Dubrovnik, Croatia. Association for Computational Linguistics.

Michel Schwab, Robert Jäschke, Frank Fischer, and Jannik Strötgen. 2019. “a buster keaton of linguistics”: First automated approaches for the extraction of vossian antonomasia. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6238–6243, Hong Kong, China. Association for Computational Linguistics.

Michel Schwab, Robert Jäschke, and Frank Fischer. 2023b. Annotated vossian antonomasia dataset.

Chuandong Su, Xiaoxi Huang, Fumiyo Fukumoto, Jiyi Li, Rongbo Wang, and Zhiqun Chen. 2020. English and chinese neural metonymy recognition based on semantic priority interruption theory. *IEEE Access*, 8:30060–30068.

Andrés Torres Rivera, Antoni Oliver, Salvador Climent, and Marta Coll-Florit. 2020. Neural metaphor detection with a residual biLSTM-CRF model. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 197–203, Online. Association for Computational Linguistics.

Christian Trippe. 2005. "change is needed to get germany moving". Accessed on 06 23, 2023.

Shun Wang, Yucheng Li, Chenghua Lin, Loic Barrault, and Frank Guerin. 2023. Metaphor detection with effective context denoising. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1404–1409, Dubrovnik, Croatia. Association for Computational Linguistics.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.

Yorick Wilks. 1978. Making preferences more active. *Artificial intelligence*, 11(3):197–223.

A Appendix

Hyperparameter optimization We conducted hyperparameter optimization using grid search on all models using F_1 score and a validation dataset that is 1/4 of the proposed test data, as explained in Section 3. The hyperparameters were tuned over the values given in Table 6. The values that we finally used for our models are given in Table 7.

hyperparameter	tested values
number of epochs	2, 3, 4, 5
batch size	8, 16, 32
maximal length	32, 64, 128
learning rate	$10^{-5}, 3 \cdot 10^{-5}, 5 \cdot 10^{-5}$
dropout rate	0.1, 0.2

Table 6: Values used for hyperparameter optimization.

model	epochs	batch size	max length	learning rate	dropout
BERT_SEQ	2	16	64	10^{-5}	0.2
BERT_MASK	4	16	32	$3 \cdot 10^{-5}$	0.2
RoBERTa_MIP	4	32	32	10^{-5}	0.2
RoBERTa_SPV	4	32	64	10^{-5}	0.2
RoBERTa_CLF	5	32	32	$3 \cdot 10^{-5}$	0.2
RoBERTA_PAIR	4	32	32	10^{-5}	0.2

Table 7: Final choice of model parameters.