One Size Fits All? A simple LSTM for Non-literal Token- and Construction-level Classification

Erik-Lân Do Dinh, Steffen Eger, and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
http://www.ukp.tu-darmstadt.de

Abstract

We tackle four different tasks of non-literal language classification: token and construction level metaphor detection, classification of idiomatic use of infinitive-verb compounds, and classification of non-literal particle verbs. One of the tasks operates on the token level, while the three other tasks classify constructions such as "hot topic" or "stehen lassen" (*to allow sth. to stand* vs. *to abandon so.*). The two metaphor detection tasks are in English, while the two non-literal language detection tasks are in German. We propose a simple context-encoding LSTM model and show that it outperforms the state-of-the-art on two tasks. Additionally, we experiment with different embeddings for the token level metaphor detection task and find that 1) their performance varies according to the genre, and 2) Mikolov et al. (2013) embeddings perform best on 3 out of 4 genres, despite being one of the simplest tested models. In summary, we present a large-scale analysis of a neural model for non-literal language classification (i) at different granularities, (ii) in different languages, (iii) over different non-literal language phenomena.

1 Introduction

Computational research of non-literal phenomena, e.g., metonymy, idiom, and prominently metaphor detection (Veale et al., 2016), has been plentiful. For metaphor detection, most works name the Conceptual Metaphor Theory (Lakoff and Johnson, 1980) as their underlying framework, in which metaphors are modeled as cognitive mappings of concepts from a source to a target domain. However, the datasets created and used in these works often follow no unified annotation guidelines (compare Steen et al. (2010) and Tsvetkov et al. (2014)), or even no disclosed guidelines at all, e.g., Heintz et al. (2013), or annotate metaphors at different levels of granularity (Steen et al., 2010; Gutierrez et al., 2016). This is also true for many works in more general non-literal language detection. Consequently, methods are seldom compared on related tasks.

Neural networks have been successfully applied to various natural language processing tasks, but few have applied them to metaphor detection (Do Dinh and Gurevych, 2016; Rei et al., 2017) or detection of non-literal and figurative language in general. In this paper, we test whether the same simple generic neural network approach is effective for four different non-literal language detection tasks: token and construction level metaphor detection, idiom classification and classification of literal and non-literal German particle verbs. We train a neural model using LSTMs to encode the context of a metaphor candidate or non-literal compound. We show that our approach outperforms existing state-of-the-art models on two tasks, while producing competitive results on another task, independent of the mode of classification (e.g., token vs. construction classification). In demonstrating the applicability of the same, simple neural network architecture to different non-literal language tasks, we lay the foundation for a more integrative approach. A joint modeling of these tasks, through data concatenation and multi-task learning, is investigated in Do Dinh et al. (2018).

Given enough training data, our model renders many of the handcrafted features employed in previous work unnecessary. This includes e.g., abstractness values to model source and target concepts (Tsvetkov

Proceedings of Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, pages 70–80
Santa Fe, New Mexico, USA, August 25, 2018.

et al., 2014; Turney and Assaf, 2011), selectional preference violations (Wilks, 1978; Shutova, 2013) or topic modeling (Heintz et al., 2013; Beigman Klebanov et al., 2014). In contrast, because they are the only external resource we utilize, we investigate the influence of an important hyper-parameter of our network—different pre-trained embeddings—on the token-level metaphor detection task and show the genre-specific effects of these embedding models.

2 Related work

Classification and detection of non-literal language has largely focused on metaphor detection. Another prominent task is the detection of idiomatic language. Similar features have been employed in those tasks, even though the specific phenomena differ. However, since the datasets used for these tasks are annotated differently, it is difficult to compare methods across the different tasks (or even subtasks of, e.g., metaphor detection). For some tasks, feature-based approaches are still superior to neural models. For many, more general, non-literal language tasks, neural models have not yet been applied. While distributed word representations have been used even in feature-based methods, a comparison regarding the influence of different pre-trained embeddings on these tasks has not been carried out so far.

Tsvetkov et al. (2014) classify adjective-noun pairs and subject-verb-object constructions. Their features include imageability, abstractness ratings, supersenses and low-dimensional word representations trained using an LSA variant. They train their system on English, and test it on four different languages—English, Russian, Farsi and Spanish—with the help of bilingual dictionaries. Similar, Gutierrez et al. (2016) examine adjective-noun compounds, specifically those for which the interaction between the components is sufficient to determine metaphoricity. To this end, they adapt a compositional distributional semantic model (CDSM) approach, representing adjectives as matrices and nouns as vectors. By computing distinct representations for literal and metaphorical use of adjectives they introduce a separation of literal and metaphorical meaning in the CDSM.

Do Dinh and Gurevych (2016) also investigate metaphor detection, however in contrast to the previously described works on a token level. Specifically, they use a multi-layer perceptron to classify metaphoric tokens using concatenated pre-trained word embeddings. Their approach is language-agnostic as it does not use additional features. However, they only test it on English data, on which it compares favorably to a simple SVM baseline and an existing feature-based method. A more complex neural model is proposed by Rei et al. (2017). They implement a similarity network in which they modulate the word representation of a token in a possibly metaphoric construction based on the remaining construction tokens. Further, they introduce a mapping from the vector space of the pre-trained embeddings to a metaphor-specific vector space. While their system performs well, it cannot beat the feature-based system by Tsvetkov et al. (2014) on adjective-noun constructions.

Zhang and Gelernter (2015) investigate metonymy identification, i.e. identification of instances where entities replace other associated entities. For example in the sentence "Washington and Beijing enter new trade talks", *Washington* and *Beijing* are used to refer to the US and Chinese governments. Zhang and Gelernter (2015) reuse many features commonly used for the metaphor detection task, such as imageability and abstractness ratings. They further test different word representations—word embeddings, LSA, and one-hot-encoding—to detect metonymy using an SVM.

A different non-literal language task is investigated by Horbach et al. (2016), in which they classify literal and idiomatic use of different German infinitive-verb compounds based on their context. They employ Naive Bayes and various features—including local skip-n-grams, POS tags, automatically obtained subject and object information, selectional preferences, and manually annotated topic information.

Köper and Schulte im Walde (2016) classify literally and non-literally used German particle verbs across 10 particles. Using a random forest classifier and various features (e.g., unigrams, affective ratings, distributed word representations), they achieve an accuracy of 85% over all particle verbs, and find that taking into account particle information additionally increases performance.

Task	dataset	size	M	lang	example
Token level metaphor detection	VUAMC	103,865	15%	en	Along with Sir James he **found** the US much more **attractive**, [...]
Construction level metaphor detection	Tsvetkov et al. (2014)	1,738	47%	en	Wind and wave power providing the **green energy** of the future.
Classification of idiomatically used verb compounds	Horbach et al. (2016)	5,249	64%	de	„Auch eine Uhr, die **stehen geblieben** ist, geht zweimal am Tag richtig", sagt er. (“A clock that has **stopped running** is correct two times a day, too,” he says.)
Classification of non- literally used particle verbs	Köper and Schulte im Walde (2016)	6,436	35%	de	Auf Decken sitzt man ums Feuer und lässt den ereignisreichen Tag **nachklingen**. (One sits on blankets around the fire and lets the day **linger on** [lit.: ring on].)

Table 1: Investigated tasks and datasets. *Size* describes labeled tokens in case of token level metaphor detection (content tokens), and labeled constructions for the other tasks respectively, *M* denotes percentage of non-literal labels. Non-literal use of tokens/constructions in the examples is marked bold.

3 Tasks

To investigate if our generic network can successfully tackle different non-literal language detection tasks, we consider token level metaphor detection, construction level metaphor detection, classification of idiomatically used infinitive-verb compounds, and classification of particle verbs into literal and non-literal usage. Table 1 gives an overview along with examples. The corpora differ in size, percentage of non-literal instances, and language.

For **token-level metaphor detection** we use the VU Amsterdam metaphor corpus (VUAMC) (Steen et al., 2010), a subset of the BNC Baby covering four genres: *academia*, *conversation*, *fiction*, and *news*. Metaphors are annotated on a token level using MIPVU (Steen et al., 2010), which in short specifies that all tokens which are not used in their most basic (concrete, bodily-related, or historically older) sense are to be labeled as metaphorical if the contextual meaning of the token can be understood in comparison with its basic sense. Inter-annotator agreement of 0.84 Fleiss' κ has been reported for this dataset. Our network is trained on each genre of the VUAMC separately; for each subcorpus we use a random subset of 76% of the data for training, 12% as a development set and 12% as a test set, reproducing the experimental setup of Do Dinh and Gurevych (2016). We re-implement their state-of-the-art approach on this dataset to compare both architectures.

We also examine **metaphor detection on construction level** utilizing the English data set created by Tsvetkov et al. (2014), specifically the literal and metaphorical adjective-noun (AN) samples, which were also used in the neural model by Rei et al. (2017). Originally these were explicitly selected for their context-independence (i.e., they should be distinguishable as metaphorical or literal based on the construction's tokens, without help from their context). We augment the published training set (i.e., the constructions) by randomly selecting for each construction a sentence containing it from the British National Corpus (BNC Consortium, 2007) and ukWac (Baroni et al., 2009). In this way, we attain 1538 sentences in total, on which we train using 10-fold cross validation. For testing, we use the 200 sentences from the original test set, which was labeled by 5 annotators achieving a Fleiss' κ of 0.74. The metaphor definition is broader than for the VUAMC; the annotators where asked to mark all tokens which “in your opinion, are used non-literally in the following sentences.”

Idiom classification is the task of deciding whether a given phrase is used idiomatically or literally. As a figurative language classification problem, and because determining whether a given phrase is used

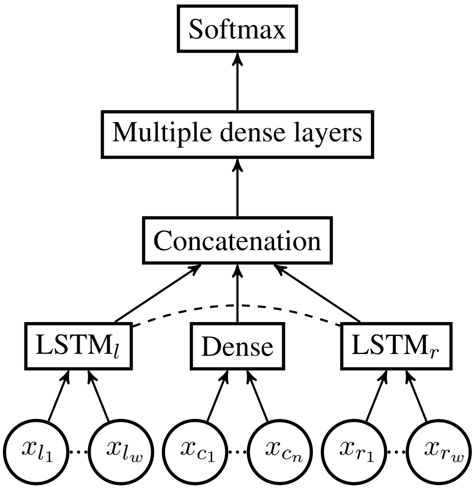

Figure 1: The basic structure of our LSTM, starting with pre-trained word embeddings x_i. Hyper-parameters include number of dense layers before applying softmax, whether or not the same LSTM layer is being used for encoding the context (i.e., $\text{LSTM}_l = \text{LSTM}_r$), and layer sizes. Center size n is determined by the corpus (e.g., $n = 1$ for token classification or $n = 2$ for adjective-noun classification).

idiomatically is largely context-dependent, this task is closely related to metaphor detection. We use the corpus introduced by Horbach et al. (2016), comprising 5,249 sentences containing literal and idiomatic uses of 6 different German infinitive-verb compounds, stemming from the Wahrig corpus (Krome (2010), covering newspaper and magazine articles). Cohen's κ for two expert annotators is reported to range between 0.63 and 0.87, with no explicit guidelines set other than to annotate the compounds as being used literally or idiomatically in a given context sentence. We run experiments on the 6 compounds separately and set up the data in a similar way as Horbach et al. (2016), i.e., we directly report accuracy scores on 10-fold cross validation experiments (averaged over 50 randomly sampled hyper-parameter configurations), without using a separate test set.

We also evaluate our approach on another task using German data: **classification of German particle verbs** into literal and non-literal cases, proposed by Köper and Schulte im Walde (2016). They compiled a corpus using 159 German particle verbs across 10 particles, extracting up to 50 sentences for each particle verb from DECOW14AX (Schäfer and Bildhauer, 2012; Schäfer, 2015). Annotators were asked to label instances on a 6-point scale from "clearly literal" to "clearly non-literal". Inter-annotator agreement of the binarized labels is reported as 0.70 Fleiss' κ for 3 annotators. We use the same setup as in the original paper and perform cross validation on the complete dataset; again we report average accuracy and F_1 over 50 randomly sampled hyper-parameter configurations. We compare our results to their follow-up paper (Köper and Schulte im Walde, 2017), in which they investigate multi-sense embeddings for this task.

4 Architecture

Our approach separately encodes the context of potential metaphorical tokens or constructions (Figure 1). More specifically, our neural network is designed to encode the left and right context of tokens/constructions using Long-Short Term Memory layers (LSTMs), to reduce the influence of the context tokens compared to just concatenating their corresponding word embeddings. This design decision stems from preliminary experiments in which we included the complete sentence context. However, this amount of context was too large to obtain reasonable results. Still, we encode the context using LSTMs rather than fully connected layers, since this provides a more concise model with fewer parameters. The *center* consists of one or two embeddings (depending on the task), which are encoded using a fully connected layer. The output of left context LSTM, center dense layer, and right context LSTM are then concatenated, before being fed to additional fully connected layers. We experiment with different network variations: shared/separate embedding layers (with re-trainable embeddings), shared/separate weights

	Token level metaphor detection		Construction level metaphor detection		Classification of idiomatically used infinitive-verb compounds		Classification of non-literally used particle verbs	
	D	LSTM	R	LSTM	H	LSTM	K	LSTM
A	**0.87**	0.86	**0.83**	0.81	0.86	**0.89**	–	0.89
F_1	0.56	**0.59**	**0.81**	0.79	–	0.90	**0.88**	0.85

Table 2: Accuracy (A) and F_1-score of existing methods (D = Do Dinh and Gurevych (2016), R = Rei et al. (2017), H = Horbach et al. (2016), K = Köper and Schulte im Walde (2017)) and LSTM on the four investigated tasks. For both metaphor detection tasks, these are results on the test set of the best systems as determined by dev set (token level) or cross validation (construction level); for the classification of idiomatically used verb-compounds and the classification of non-literally used particle verbs these are averages over 50 configurations, since the original papers only report performance on cross validation. For subcorpus specific results see Table 3 (token level metaphor detection) and Table 4 (classification of idiomatically used infinitive-verb compounds).

for the context-LSTMs, different context representation sizes, and differing number and size of the fully-connected layers.

We adapt the input for our network to each corpus, since the context can differ depending on the task. For example, for the infinitive-verb classification, the annotated instance can consist of two tokens, thus we can have two *center* embeddings. To illustrate, consider the example:

> “Kinder sollten nicht mehr **sitzen bleiben** müssen, sondern gefördert werden.”

In this sentence, we use (*Kinder,sollten,nicht,mehr*) as left context, (*sitzen,bleiben*) as center, and (*müssen,sondern,gefördert,werden*) as right context (see Figure 1).

For the tasks with German data we use the word embeddings of Reimers et al. (2014). For construction level metaphor detection we employ the embeddings of Komninos and Manandhar (2016) as preliminary cross validation experiments on the training set show that they work well. On the other hand, preliminary experiments on the development set for token level metaphor detection show an advantage of the Google News word2vec embeddings (Mikolov et al., 2013) for this task, which is why we use them to work on the VUAMC (a more in-depth analysis validates our decision, Section 6). We conduct our experiments using Keras[1] and Theano[2]. We make our code publicly available[3].

5 Results

The main results are laid out in Table 2, results broken down into subcorpora are shown in Table 3 (token level metaphor detection) and Table 4 (classification of particle verbs). We see that our LSTM model outperforms the existing approaches on both token-level metaphor detection and classification of idiomatically used infinitive-verb compounds. The results on the remaining two tasks are slightly below the state-of-the-art, but are comparable despite not using handcrafted features. The results for the two German tasks are generally higher for all approaches, because multiple instances—both for non-literal and literal use—of each construction are available in these datasets. Further, they only consider few given terms/phrases. In contrast, the English datasets provide annotations for many more different tokens (or constructions). For a closer analysis, we look into the best system for each task.

5.1 Token level metaphor detection

For the token level metaphor detection task, our LSTM yields better results on all subcorpora compared to the re-implemented MLP approach (Table 3). This is not only a matter of choosing the right hyper-parameter combination, as displayed in the much larger spread for the MLP (an example shown for the *news* subcorpus in Figure 2), which is similarly large for both *academia* and *fiction* subcorpora.

[1] v2.0.0, `github.com/fchollet/keras`

[2] v0.9.0, `deeplearning.net/software/theano`

[3] `https://github.com/UKPLab/latech-cflf-2018-nonliteral`

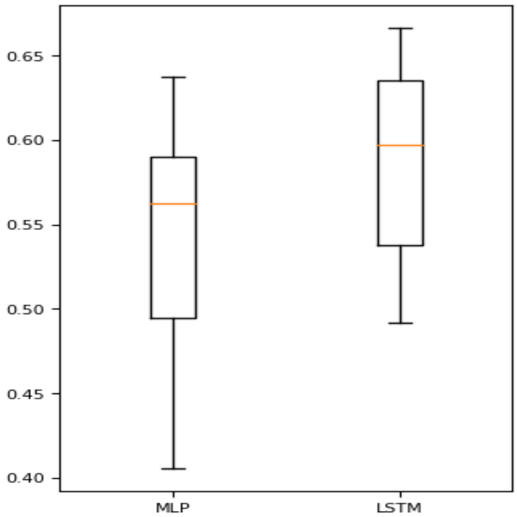

Figure 2: **Token level metaphor detection**. Whisker plot comparing MLP and LSTM configurations performance spread according to development set F$_1$-score on the *news* subcorpus of the VUAMC. Whiskers extend to 1.5 interquartile range below the first and above the third quartile.

	Do Dinh and Gurevych (2016)	LSTM
academia	0.5916	**0.6341**
conversation	0.5023	**0.5410**
fiction	0.5259	**0.5555**
news	0.6251	**0.6448**
mean	0.5612	**0.5939**

Table 3: **Token level metaphor detection**. F$_1$-score of the MLP and the LSTM on the genre-specific VUAMC test sets.

In contrast, both networks show comparably high variance for the *conversation* subcorpus. We attribute this to the often short sentence context in this subcorpus. For example, in 1/3 of the 159 instances in which both MLP and LSTM wrongly classify a token, sentence length is shorter than 9 tokens. This is only the case for roughly 1/9 of the correctly classified tokens. However, theoretically sufficient length does not guarantee sufficient context. Consider, e.g., the sentence

> “you **see** put John, one of us start trussing early [...]” (metaphor in bold),

where ungrammaticality, omissions, and missing wider context make classification difficult, even for humans. This is true to a lesser extent also for the remaining corpus.

Investigating specific word forms indicates that in some cases the networks do not learn enough from the context. For example, 64 (out of 209) instances of the verb form “see” are annotated as being metaphorical in the training set, but the MLP seems unable to incorporate this information, labeling only 1 instance in the test set as metaphorical. In contrast, the LSTM labels only 1 instance of “take” as literal, even though the training set contains 50 (out of 119) literal examples.

5.2 Construction level metaphor detection

In this task, our generic model is slightly outperformed by a more complex task-tailored network (Rei et al., 2017). The original feature-based approach (Tsvetkov et al. (2014), F$_1$-score of 0.85) is still not in reach for both neural network approaches. However, since the original test set is quite small (200 instances), the results of all those approaches have to be interpreted carefully.

Our approach yields considerably lower recall (0.720) than precision (0.878) for this dataset. Ten constructions are wrongly labeled as metaphorical. Of those, four contain adjectives that are also part of wrongly literally labeled constructions: *honest opinion, unruly behavior, cool [dry] air, Clear [blue] skies* are wrongly labeled metaphorical. In contrast, *honest meal, unruly hair, cool feature, clear explanations* are misclassified as literal. Looking at nouns in the constructions, we observe that all pairs containing “voice” (*silky voice* (M), *shrill voices* (L), *quiet voice* (L)) are labeled as metaphor, while all the instances containing “brain” (*foggy brain*, *rusty brain*) are mislabeled as being literally used.

These examples illustrate that for construction level metaphor detection the interaction between the construction components is more important than the remaining context. Also, misclassified constructions appear at the beginning or end only in 24% of their containing sentences, compared to 28% of the correctly labeled ones. This further confirms that larger context is less important for this task than the immediate interaction between adjective and noun. Since we do not model this interaction explicitly, our network is outperformed by the approach of Tsvetkov et al. (2014) on the sparse amount of training data. However, even without this explicit modeling, our simple neural approach performs nearly as well as the much more specialized approach of Rei et al. (2017) which models this interaction specifically.

5.3 Classification of idiomatically used infinitive-verb compounds

For this task, we not only outperform the approach of Horbach et al. (2016) averaged over all infinitive-verb compounds, but for each individual compound. This is most pronounced for "hängen bleiben" and "liegen bleiben".

We examine the compounds on which our network performs best ("sitzen lassen" – *leave sitting*) and worst ("stehen bleiben" – *stay standing*) on average (Table 4). For "stehen bleiben" we see that for 48% of the instances which the LSTM mislabels the compound appears at the end of the sentence, meaning that basically no right context is available. This is only the case for 44% of the correctly labeled instances, indicating that further hyper-parameter optimization without changing the architecture can only increase performance to a degree. "sitzen lassen" has a highly skewed label distribution of only 44 of 881 instances being annotated as literal. The large number of false positive classifications in relation to actual literal instances (18 of the 44 literal instances are classified as non-literal) thus has only negligible impact on precision and F_1-score. 2/3 of those false positives contain the construction directly or very near the end of the sentence, highlighting again the problem with the windowed approach.

	Horbach et al. (2016)	LSTM
hängen+bleiben	0.836	**0.875**
liegen+bleiben	0.847	**0.881**
sitzen+bleiben	0.875	**0.904**
sitzen+lassen	0.946	**0.970**
stehen+bleiben	0.812	**0.817**
stehen+lassen	0.847	**0.861**
average	0.861	**0.885**

Table 4: **Classification of idiomatically used infinitive-verb compounds**. Accuracy values for Horbach et al. (2016) and LSTM (averaged over 50 configurations).

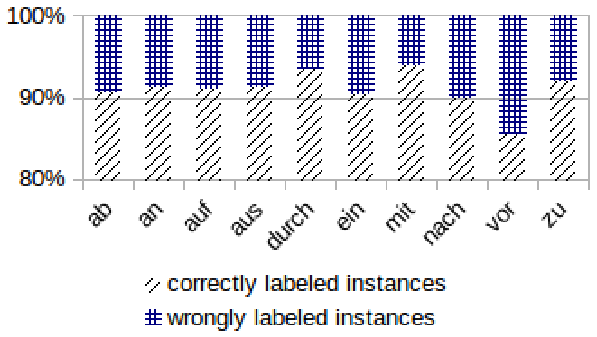

Figure 3: **Particle verb classification**. Accuracy across the ten different particles. Average macro-accuracy is 90.4%.

5.4 Classification of non-literal particle verbs

Classification error rates for the non-literal particle verbs are similar across the particles. Figure 3 shows that three particles stand out, namely: "durch" and "mit" exhibit a far lower error rate (6.8% and 6.3% respectively) and the particle "vor" higher (16.7%) than average (9.6%).

The corpus contains instances for two verbs that start with the particle "vor". While "vordrängen" (*to press forward*) is represented by mainly literal instances (91%), the distribution for "vorschalten" (*to prepose*) is rather balanced (literal: 46%). However, our model produces more classification errors for the former. For the particle "zu", "zustopfen" (*to plug*) is the only verb which also shows a fairly balanced label distribution. However, it also is responsible for almost half of the misclassifications made for verbs with "zu". Other balanced verbs show again different behavior; e.g., "einbrechen" (*to break in*) is misclassified only in 4 of 44 instances. We see that the amount of literal or non-literal training instances for one particular verb is not the deciding factor for classification accuracy. Instead, the network apparently manages to abstract over verbs, however, also introduces some errors in the process.

Embeddings	training data	type / method	coverage
word2vec (Mikolov et al., 2013)	Google News texts	skip-gram with negative sampling	87.6%
GloVe (Pennington et al., 2014)	Wikipedia, newswire	word-word co-occurrence statistics	92.0%
Conceptnet Numberbatch (Speer et al., 2017)	word2vec, GloVe, knowledge bases	combination of existing embeddings and knowledgebases using retrofitting	84.8%
Levy and Goldberg (2014)	Wikipedia	dependency-based	87.8%
Komninos and Manandhar (2016)	Wikipedia	dependency-based and token windows	89.6%

Table 5: Embeddings tested with classification, and their coverage of the VUAMC.

6 Effects of different embeddings

Next, we analyze more closely the effects of a special hyper-parameter on the detection of non-literal language: the pre-trained word embeddings used in our network. Our intuition is that embeddings trained on a similar domain as the test data lead to better results. To test this, we replicate the token level metaphor detection experiments using different word embedding models and sample ten hyper-parameter configurations, from which we choose the best performing (development set) respectively. We use the pre-trained embeddings detailed in Table 5 (all have 300 dimensions).

Coverage, i.e., how many of the tokens in the corpus have an embedding representation, only has a minimal effect on performance. This is illustrated, e.g., by the Glove embeddings, which have the highest coverage but by far the worst overall performance. Recall from Table 3 that metaphors from the *conversation* and *fiction* genres seem to be harder to detect in general, owing to larger context dependence, higher ambiguity, and in case of *conversation* to fragmented sentences and omission. Indeed, we find that *conversation* and *fiction* texts exhibit the largest differences and the worst results regardless of embeddings used. We note that arguably, those genres are the most different from the news texts and Wikipedia articles that the embeddings are trained on. Independently of the concrete embeddings used, the network performs consistently best on the *news* subcorpus, followed by the *academic* texts.

Looking more closely into the classifications on the *fiction* subcorpus, we observe a large performance difference between Glove and the remaining embeddings. This is mainly due to low recall (0.356, see also Table 6), especially compared to the word2vec embeddings (0.710). The results on the *conversation* subcorpus are similarly noteworthy, because here both embedding models that encode dependency information, from Levy and Goldberg (2014) and Komninos and Manandhar (2016), perform worse than the remaining models (also due to lower recall). This is in line with our findings from Section 5.1 where we note that our network struggles with omissions or ungrammatical sentences—as the structure of the conversation sentences is more likely to be irregular, including “correct” syntactic information can apparently be detrimental.

	academic			conversation			fiction			news			avg
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	F_1
word2vec	.576	**.706**	**.634**	.567	.518	**.541**	.456	**.710**	**.555**	**.640**	.650	.645	**.592**
GloVe	.544	.594	.568	.470	**.584**	.521	.553	.356	.433	.598	.580	.589	.504
ConceptNet	.604	.654	.628	.595	.478	.530	**.570**	.486	.524	.621	.706	**.661**	.576
Levy	**.652**	.535	.588	.629	.439	.517	.485	.545	.513	.636	.645	.640	.553
Komninos	.634	.628	.631	**.652**	.396	.493	.511	.587	.547	.601	**.712**	.652	.569

Table 6: System precision (P), recall (R), and F_1-score for the VUAMC using different embeddings.

At the end, while e.g., the embeddings by Komninos and Manandhar (2016) perform close to the word2vec embeddings in most genres, the fact that they perform relatively poorly on the *conversation* transcripts make them a bad fit for general metaphor identification. The Conceptnet embeddings show better performance on the news subcorpus, however this is no substantial improvement over the generally better performing word2vec embeddings—which do not rely on further knowledgebases.

7 Conclusion

We conducted a large scale study on distinguishing literal from non-literal language on four different tasks, using a generic neural network. These tasks were: token level metaphor detection, construction level metaphor detection, classification of idiomatically used infinitive-verb compounds, and classification of literally or non-literally used particle verbs. Our tasks comprised two languages: English and German. We find that, while the tasks differ with regards to annotation scheme and supposed context dependence, and their respective datasets differ with regards to size and label balance, our generic simple neural model outperforms existing state-of-the-art models on two of four tasks using only pre-trained embeddings, and on the remaining tasks produces competitive results to more task-tailored or feature-based approaches.

Further, we investigated the influence of different pre-trained word embeddings for one of the tasks, token level metaphor classification. We find that performance depends less on the underlying genre than on the architecture used.

In future work, we want to explore how commonalities between the investigated and similar tasks can be exploited, e.g., using multi-task learning (Collobert and Weston, 2008), where we not only share the architecture, but also the parameters of the network among the investigated tasks.

Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 01UG1816B (CEDIFOR).

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Beata Beigman Klebanov, Chee Wee Leong, Michael Heilman, and Michael Flor. 2014. Different Texts, Same Metaphors: Unigrams and Beyond. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 11–17, Baltimore, MD, USA. Association for Computational Linguistics.

The BNC Consortium. 2007. The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of ICML 2008*, pages 160–167, Helsinki, Finland. ACM.

Erik-Lân Do Dinh and Iryna Gurevych. 2016. Token-Level Metaphor Detection using Neural Networks. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 28–33, San Diego, CA, USA. Association for Computational Linguistics.

Erik-Lân Do Dinh, Steffen Eger, and Iryna Gurevych. 2018. Killing Four Birds with Two Stones: Multi-Task Learning for Non-Literal Language Detection. In *Proceedings of COLING 2018*, page (to appear), Santa Fe, NM, USA. ICCL.

Elkin Dario Gutierrez, Ekaterina Shutova, Tyler Marghetis, and Benjamin Bergen. 2016. Literal and Metaphorical Senses in Compositional Distributional Semantic Models. In *Proceedings of ACL 2016*, pages 183–193, Berlin, Germany. Association for Computational Linguistics.

Ilana Heintz, Ryan Gabbard, Donald S Black, Marjorie Freedman, Ralph Weischedel, and San Diego. 2013. Automatic Extraction of Linguistic Metaphor with LDA Topic Modeling. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 58–66, Atlanta, GA, USA. Association for Computational Linguistics.

Andrea Horbach, Andrea Hensler, Sabine Krome, Jakob Prange, Werner Scholze-Stubenrecht, Diana Steffen, Stefan Thater, Christian Wellner, and Manfred Pinkal. 2016. A Corpus of Literal and Idiomatic Uses of German Infinitive-Verb Compounds. In *Proceedings of LREC 2016*, pages 836–841, Portorož, Slovenia. European Language Resources Association.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency Based Embeddings for Sentence Classification Tasks. In *Proceedings of NAACL-HLT 2016*, pages 1490–1500, San Diego, CA, USA. Association for Computational Linguistics.

Sabine Krome. 2010. Die deutsche Gegenwartssprache im Fokus korpusbasierter Lexikographie. Korpora als Grundlage moderner allgemeinsprachlicher Wörterbücher am Beispiel des Wahrig Textkorpus digital. In Iva Kratochvílová and Norbert Richard Wolf, editors, *Kompendium Korpuslinguistik. Eine Bestandsaufnahme aus deutsch-tschechischer Perspektive*, pages 117–134. Heidelberg: Universitätsverlag Winter.

Maximilian Köper and Sabine Schulte im Walde. 2016. Distinguishing Literal and Non-Literal Usage of German Particle Verbs. In *Proceedings NAACL-HLT 2016*, pages 353–362, San Diego, CA, USA. Association for Computational Linguistics.

Maximilian Köper and Sabine Schulte im Walde. 2017. Applying Multi-Sense Embeddings for German Verbs to Determine Semantic Relatedness and to Detect Non-Literal Language. In *Proceedings of EACL 2017*, pages 535–542, Valencia, Spain. Association for Computational Linguistics.

George Lakoff and Mark Johnson. 1980. *Metaphors we live by*. University of Chicago Press, Chicago, IL, USA.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of ACL 2014*, pages 302–308, Baltimore, MD, USA. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS 2013*, pages 3111–3119, Lake Tahoe, NV, USA. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP 2014*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the Finer Point : A Supervised Similarity Network for Metaphor Detection. In *Proceedings of EMNLP 2017*, pages 1538–1547, Copenhagen, Denmark. Association for Computational Linguistics.

Nils Reimers, Judith Eckle-Kohler, Carsten Schnober, Jungi Kim, and Iryna Gurevych. 2014. GermEval-2014: Nested Named Entity Recognition with Neural Networks. In *Workshop Proceedings of the 12th Edition of the KONVENS Conference*, pages 117–120, Hildesheim, Germany. Universitätsverlag Hildesheim.

Roland Schäfer and Felix Bildhauer. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of LREC 2012*, pages 486–493, Istanbul, Turkey. ELRA.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, pages 28–34, Lancaster, UK. Institut für Deutsche Sprache.

Ekaterina Shutova. 2013. Metaphor Identification as Interpretation. In *Proceedings of *SEM*, pages 276–285, Atlanta, GA, USA. Association for Computational Linguistics.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of AAAI 2017*, pages 4444–4451, San Francisco, CA, USA. AAAI Press.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification. From MIP to MIPVU*. John Benjamins, Amsterdam, Netherlands.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor Detection with Cross-Lingual Model Transfer. In *Proceedings of ACL 2014*, pages 248–258, Baltimore, MD, USA. Association for Computational Linguistics.

Peter D Turney and Dan Assaf. 2011. Literal and Metaphorical Sense Identification through Concrete and Abstract Context. In *Proceedings of EMNLP 2011*, pages 680–690, Edinburgh, United Kingdom. Association for Computational Linguistics.

Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov. 2016. *Metaphor: A Computational Perspective. Synthesis Lectures on Human Language Technologies.* Morgan & Claypool, USA.

Yorick Wilks. 1978. Making Preferences More Active. *Artificial Intelligence*, 11(3):197–223.

Wei Zhang and Judith Gelernter. 2015. Exploring Metaphorical Senses and Word Representations for Identifying Metonyms. *arXiv preprint, arXiv:1508.04515*.