Supervised Word-Level Metaphor Detection: Experiments with Concreteness and Reweighting of Examples

Beata Beigman Klebanov, Chee Wee Leong, and Michael Flor
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541
bbeigmanklebanov,cleong,mflor@ets.org

Abstract

We present a supervised machine learning system for word-level classification of all content words in a running text as being metaphorical or non-metaphorical. The system provides a substantial improvement upon a previously published baseline, using re-weighting of the training examples and using features derived from a concreteness database. We observe that while the first manipulation was very effective, the second was only slightly so. Possible reasons for these observations are discussed.

1 Introduction

In this paper, we present a set of experiments aimed at improving on previous work on the task of supervised word-level detection of linguistic metaphor in running text. The use of supervised machine learning techniques for metaphor identification has increased manyfold in the recent years (see section 10, Related Work, for a review and references), partially due to the availability of large-scale annotated resources for training and evaluating the algorithms, such as the VU Amsterdam corpus (Steen et al., 2010), datasets built as part of a U.S. government-funded initiative to advance the state-of-art in metaphor identification and interpretation (Mohler et al., 2013; Strzalkowski et al., 2013), and recent annotation efforts with other kinds of data (Beigman Klebanov and Flor, 2013; Jang et al., 2014). Some of these data are publicly available (Steen et al., 2010), allowing for benchmarking and for measuring incremental improvements, which is the approach taken in this paper.

Data	#Texts	content tokens	% metaphors
News	49	18,519	18%
Fiction	11	17,836	14%
Academic	12	29,469	13%
Conversation	18	15,667	7%
Essay Set A	85	21,838	11%
Essay Set B	79	22,662	12 %

Table 1: The sizes of the datasets used in this study, and the proportion of metaphors. Content tokens are nouns, adjectives, adverbs, and verbs.

We start with a baseline set of features and training regime from Beigman Klebanov et al. (2014), and investigate the impact of re-weighting of training examples and of a suite of features related to concreteness of the target concept, as well as to the difference in concreteness within certain types of dependency relations. The usage of concreteness features was previously discussed in the literature; to our knowledge, these features have not yet been evaluated for their impact in a comprehensive system for word-level metaphor detection, apart from the concreteness features as used in Beigman Klebanov et al. (2014), which we use as a baseline.

2 Data

2.1 VU Amsterdam Data

We use the VU Amsterdam metaphor-annotated dataset.[1] The dataset consists of fragments sampled across four genres from the British National

[1] http://www2.let.vu.nl/oz/metaphorlab/metcor/search/index.html

Proceedings of the Third Workshop on Metaphor in NLP, pages 11–20,
Denver, Colorado, June 5, 2015.

Corpus (BNC): Academic, News, Conversation, and Fiction. The data is annotated according to the MIPVU procedure (Steen et al., 2010) with the inter-annotator reliability of $\kappa > 0.8$.

In order to allow for direct comparison with prior work, we used the same subset of these data as Beigman Klebanov et al. (2014), in the same cross-validation setting. The total of 90 fragments are used in cross-validation: 10-fold on News, 9-fold on Conversation, 11 on Fiction, and 12 on Academic. All instances from the same text were always placed in the same fold. Table 1 shows the sizes of the datasets for each genre, as well as the proportion of metaphors therein.

2.2 Essay Data

The dataset contains 174 essays written for a large-scale college-level assessment of analytical writing. The essays were written in response to one of the following two topics: Discuss the statement "High-speed electronic communications media, such as electronic mail and television, tend to prevent meaningful and thoughtful communication" (Set A, 85 essays), and "In the age of television, reading books is not as important as it once was. People can learn as much by watching television as they can by reading books." (Set B, 79 essays). These essays were annotated for argumentation-relevant metaphors (Beigman Klebanov and Flor, 2013), with inter-annotator reliability of κ = 0.58 and κ = 0.56 for Set A and Set B, respectively. We will report results for 10-fold cross-validation on each of sets A and B, as well as across prompts, where the machine learner would be trained on Set A and tested on Set B and vice versa. Please refer to Table 1 for further details about the datasets. This dataset was used in Beigman Klebanov et al. (2014), allowing for a direct comparison.

3 Experimental Set-Up

In this study, each content-word token in a text is an instance that is classified as either a metaphor or not a metaphor. We use the logistic regression classifier as implemented in the SKLL package (Blanchard et al., 2013), which is based on scikit-learn (Pedregosa et al., 2011), with F1 optimization ("metaphor" class). Performance will be evaluated using Precision, Recall, and F-1 score, for the positive ("metaphor") class.

As a baseline, we use the best performing feature set from Beigman Klebanov et al. (2014), who investigated supervised word-level identification of metaphors. We investigate the effect of reweighting of examples, as well as the effectiveness of features related to the notion of concreteness.

4 Baseline System

As a baseline, we use the best feature set from Beigman Klebanov et al. (2014). Specifically, the baseline contains the following families of features:

- Unigrams;
- Part-of-speech tags generated by Stanford POS tagger 3.3.0 (Toutanova et al., 2003);
- Mean concreteness values from Brysbaert et al. (2013) set of concreteness norms, represented using 0.25-wide bins that span the 1-5 range of possible values;
- $log\frac{P(w|t)}{P(w)}$ values for each of 100 topics generated by Latent Dirichlet Allocation (Blei et al., 2003) from the NYT corpus (Sandhaus, 2008).

5 Experiment 1: Re-weighting of Examples

Given that the category distribution is generally heavily skewed towards the non-metaphor category (see Table 1), we experimented with cost-sensitive machine learning techniques to try to correct for the imbalanced class distribution (Yang et al., 2014; Muller et al., 2014). The first technique uses **AutoWeight** (as implemented in the *auto* flag in scikit-learn toolkit), where we assign weights that are inversely proportional to the class frequencies.[2] Table 2 shows the results.

The effect of auto-weighting on the VUA data is quite dramatic: A 14-point drop in precision is offset by a 32-point increase in recall, on average, along with a 10-point average increase in F1 score. The precision-recall balance for VUA data changed from P=0.58, R=0.34 to P=0.44, R=0.66, nearly doubling

[2]The re-weighting of examples was only applied to training data; the test data is unweighted.

Data	Baseline			AutoWeighting		
	P	R	F	P	R	F
A-B	.71	.35	.47	.52	.71	.60
B-A	.57	.49	.53	.40	.67	.50
Set A	.70	.48	.57	.50	.75	.60
Set B	.76	.59	.67	.57	.80	.67
Av. Essays	.69	.48	.56	.50	.74	.59
Acad.	.63	.35	.42	.53	.66	.56
Conv.	.50	.24	.32	.29	.69	.39
Fiction	.55	.29	.38	.41	.61	.49
News	.64	.46	.54	.53	.68	.59
Av. VUA	.58	.34	.41	.44	.66	.51

Table 2: Performance of a model with AutoWeighted training examples in comparison to the unweighted baseline, in terms of Precision (P), Recall (R), and F-1 score (F) for the positive ("metaphor") class. A-B and B-A correspond to training-testing scenarios where the system is trained on Set A and tested on Set B and vice versa, respectively. All other figures report average performance across the cross-validation folds.

the recall. The effect on essay data is such that the average drop in precision is larger than for VUA data (19 points) while the improvement in recall is smaller (26 points). The average increase in F-1 score is about 3 points, with the maximum of up to 13 F-1 points (A-B evaluation) and a 3-point drop for B-A evaluation.

Overall, this experiment shows that the feature set can support a radical change in the balance between precision and recall. When precision is a priority (as in a situation where feedback to the user is provided in the form of highlighting of the metaphorically used words, for example), it is possible to achieve nearly 70% precision, while recovering about half the metaphors. When recall is a priority (possibly when an overall per-essay metaphoricity rate is estimated and used as a feature in an essay scoring system), it is possible to recover about 3 out of every 4 metaphors, with about 50% precision. For VUA data, a similar trend is observed, with somewhat worse performance, on average, than on essay data. The performance on the VUA News and Academic data is in line with the findings for the cross-prompt generalization in the essay data, whereas Conversation and Fiction genres are more difficult for the current system.[3]

Having observed the results of the auto-weighting experiments, we conjectured that perhaps a more even balance of precision and recall can be obtained if the re-weighting gives extra weight to "metaphor" class, but not to the extent that the auto-weighting scheme does. In the second experiment, we tune the weight parameter using grid search on the training data (through a secondary 3-fold cross-validation within training data) to find the optimal weighting in terms of F-score (**OptiWeight**); the best-performing weight was then evaluated on the test data (for cross-prompt evaluations) or the test fold (cross-validations). We used the grid from 1:1 weighting up to 8:1, with increments of 0.33.

The first finding of note is that the optimal weighting for the "metaphor" class is lower than the auto-weight. For example, given that metaphors constitute 11-12% of instances in the essay data, the auto-weighting scheme for the A-B and B-A evaluations would choose the weights to be about 8:1, whereas the grid search settled on 3:1 when trained on prompt A and 3.33:1 when trained on prompt B. A similar observation pertains to the VUA data: The auto-weighting is expected to be about 4.5:1 for News data, yet the grid search settled on 4:1, on average across folds. These observations suggest that the auto-weighting scheme might not be the optimal re-weighting strategy when optimizing for F1 score with equal importance of precision and recall.

Table 3 shows the performance of the optimized weighting scheme. For VUA data, the changes in performance are generally positive albeit slight – the F1 score increases by one point for 3 out of 4 evaluations). For essay data, it is clear that the imbalance between precision and recall is substantially reduced (from the average difference between recall and precision of 0.24 for the auto-weighted scheme to the average difference of 0.08 for the optimized weights; see column *D* in the Table). The best effect was observed for the B-A evaluation (train on set B, test on set A) – a 6-point increase in preci-

[3]This could be partially explained by the fact that the samples for Fiction and Conversation contain long excerpts from the same text, so they allow for less diversity than samples in the News set, with a larger number of shorter excerpts, although performance on the Academic set is not quite in line with these observations.

Data	AutoWeight				OptiWeight			
	P	R	F	*D*	P	R	F	*D*
A-B	.52	.71	.60	*.19*	.58	.55	.57	*-.03*
B-A	.40	.67	.50	*.27*	.46	.65	.54	*.20*
A	.50	.75	.60	*.25*	.56	.66	.60	*.11*
B	.57	.80	.67	*.23*	.52	.69	.68	*.03*
Av.	.50	.74	.59	*.24*	.57	.64	.60	*.08*
Ac.	.53	.66	.56	*.14*	.52	.69	.57	*.17*
Con.	.29	.69	.39	*.39*	.32	.63	.40	*.31*
Fict.	.41	.61	.49	*.20*	.40	.66	.49	*.26*
News	.53	.68	.59	*.15*	.51	.71	.60	*.20*
Av.	.44	.66	.51	*.22*	44	.67	.51	*.24*

Table 3: Performance of a model with optimally weighted training examples in comparison to the auto-weighted scheme, in terms of Precision (P), Recall (R), F-1 score (F), and the difference between Recall and Precision (D). A-B and B-A correspond to training-testing scenarios where the system is trained on Set A and tested on Set B and vice versa, respectively. All other figures report average performance across the cross-validation folds.

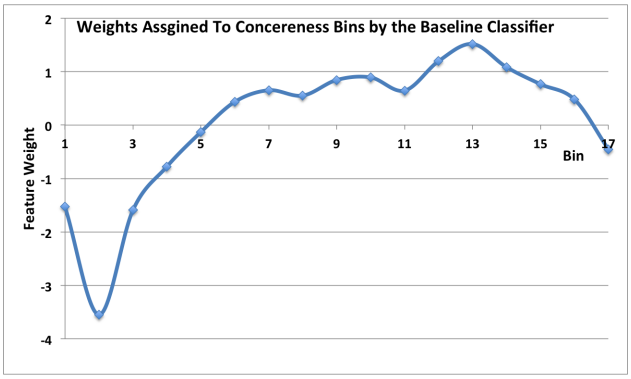

Figure 1: Weights assigned to the different concreteness bins by the logistic regression classifier with the baseline feature set in an unweighted training regime. The bins span the 1-5 range with 0.25 increments; words falling in bin 1 are the most abstract, while words falling in bin 17 are the most concrete.

sion compensated well for the 2-point drop in recall, relative to the auto-weighting scheme, with a resulting 4-point increase in F-score. The worst effect was observed for the A-B evaluation, where the increase of 6 points in precision was offset by a 16-point drop in recall. We conclude, therefore, that a grid-based optimization of weighting can help improve the precision-recall balance of the learning system and also improve the overall score in some cases.

6 Experiment 2: Re-representing concreteness information

In this paper, we use mean concreteness scores for words as published in the large-scale norming study by Brysbaert et al. (2013). The dataset has a reasonable coverage for our data; thus, 78% of tokens in Set A have a concreteness rating. The ratings are real numbers on the scale of 1 through 5; for example, *essentialness* has the concreteness of 1.04, while *sled* has the concreteness of 5.

The representation used by the baseline system bins the continuous values into 17 bins, starting with 1 and incrementing by 0.25 (the topmost bin has words with concreteness value of 5). Compared to a representation using a single continuous variable, the binned representation allows the machine-learner to provide different weights to different bins, thus modeling a non-linear relationship between concreteness and metaphoricity. Indeed, the logistic regression classifier has made precisely such use of this representation; Figure 1 shows the weights assigned by the classifier to the various bins, in a baseline model with unweighted examples trained on Set A data. Specifically, it is clear that abstract words receive a negative weight (predict the class "non-metaphor"), while concreteness values above 2.5 generally receive a positive weight (apart form the top bin, which happens to have only a single word in it).

One potential problem with binning as above is that some of the features become quite sparse; sparseness, in turn, makes them vulnerable to overfitting. Since the relationship between concreteness and feature weight is mostly monotonic (between bins 2 and 13), we experimented with defining bins that would encode various thresholds. Thus, bin $b5 = [2, 2.5]$ would fire whenever the value of the instance is at least 2 ($x \in [2, 5]$) or whenever the value of the instance is at most 2.5 ($x \in [1, 2.5]$); we call these theshold-up and threshold-down, respectively. Thus, instead of a set of 17 binary bins coding for intervals, we now have a set of 34 binary bins coding for upward and downward thresholds. The effect of this manipulation on the performance was generally small, yet this version of the concreteness feature yielded more robust performance. Specifically, the finding above of a drop in A-B performance in

the optimal-weighting scheme is now largely mitigated, with precision staying the same (0.58), while recall improving from 0.55 to 0.60, and the resulting F1 score going up from 0.57 to 0.59, just one point below the auto-weighted version. The improved performance on B-A is preserved and even further improved, with P=0.50, R=0.62, F=0.55. For the rest of the datasets and weighting regimes, the performance was within one F-score point of the performance of the baseline feature set.

7 Experiment 3: Features capturing difference in concreteness

In this section, we present results of experiments trying to incorporate contextual information about the difference in concreteness between the adjective and its head noun (**AdjN**) and between the verb and its direct object (**VN**). The intuition behind this approach is that a metaphor is often used to describe an abstract concept in more familiar, physical terms. A concrete adjective modifying an abstract noun is likely to be used metaphorically (as in *soft revolution* or *dark thought*); similarly, a concrete verb with an abstract direct object is likely to be a metaphor (as in *pour consolation* or *drive innovation*). Turney et al. (2011) introduced a method for acquiring estimates of concreteness of words automatically, and measuring difference in concreteness in AdjN and VN constructions. They reported improved metaphor classification accuracies on constructed sets of AdjN and VN pairs.

We implemented a difference-in-concreteness feature using the values from Brysbaert et al. (2013) database. We parsed texts using Stanford Dependency Parser (de Marneffe et al., 2006), and identified all instances of amod, dobj, and rcmod relations that connect an adjective to a noun (amod), a verb to its direct object (dobj), and a verb in a relative clause to its head noun (rcmod). For example, in the sentence "I read the wonderful book that you recommended," the following pairs would be extracted: *wonderful-book* (amod), *read-book* (dobj), and *recommended-book* (rcmod). The difference-in-concreteness features are calculated for the adjectives and the verbs participating in the above constructions, as follows. Let (adj,n) be a pair of words in the amod relation; then the value of the difference in concreteness (DC) for the adjective is given by:

$$DC(adj) = Concr(adj) - Concr(n) \quad (1)$$

DC(v) for pairs (v,n) in dobj or rcmod relations is defined analogously. Features based on DC apply only to adjectives and verbs participating in the eligible constructions specified above.

To represent the difference in concreteness information for the machine learner, we utilize the binned thresholded representation introduced in section 6. The range of the values is now [-4,4]; hence we define 33 bins for each of the threshold-up and threshold-down versions.

Data	UPT+ CUpDown			UPT+ CUpDown+ DCUpDown		
	P	R	F	P	R	F
A-B	.712	.355	.474	.712	.362	.480
B-A	.563	.495	.527	.565	.494	.527
Set A	.703	.478	.567	.699	.475	.564
Set B	.757	.594	.665	.760	.604	.672
Av.	.684	.481	.558	.684	.484	.561
Acad.	.633	.350	.419	.636	.356	.425
Conv.	.500	.242	.317	.487	.236	.309
Fiction	.550	.291	.377	.559	.309	.395
News	.640	.465	.536	.636	.466	.536
Av.	.581	.337	.412	.580	.342	.416

Table 4: Performance of a model trained with unweighted examples with and without DC (difference in concreteness) features.

Table 4 shows the incremental improvement as a result of adding the DCUpDown features to the system with UPT+CUpDown. The improvement in recall and in F-score is very small – up to 0.4 F1 points on average across the evaluations. The largest increase in performance is observed for the VUA Fiction data (1.8 F1 points), with increases in both precision and recall. Since unweighted training scenario generally leads to high-precision low-recall models, an improvement in recall without drop in precision is helping the system to achieve a more balanced performance.

Table 5 shows the incremental improvements in performance when the system is trained in the auto-

Data	UPT+ CUpDown			UPT+ CUpDown+ DCUpDown		
	P	R	F	P	R	F
A-B	.521	.716	.603	.528	.713	.607
B-A	.401	.672	.503	.415	.670	.513
Set A	.499	.751	.597	.500	.747	.597
Set B	.571	.792	.663	.592	.773	.669
Av.	.498	.733	.592	.509	.726	.597
Acad.	.525	.662	.564	.525	.657	.562
Conv.	.292	.691	.393	.293	.691	.396
Fiction	.408	.608	.485	.411	.607	.486
News	.528	.674	.590	.530	.673	.590
Av.	.438	.659	.508	.440	.657	.509

Table 5: Performance of a model trained with auto-weighted examples with and without DC (difference in concreteness) features.

Data	UPT+ CUpDown			UPT+ CUpDown+ DCUpDown		
	P	R	F	P	R	F
A-B	.584	.596	.590	.593	.556	.574
B-A	.499	.620	.553	.485	.635	.550
Set A	.562	.659	.603	.561	.661	.604
Set B	.674	.697	.684	.662	.722	.690
Av.	.580	.643	.608	.575	.644	.605
Acad.	.532	.655	.564	.531	.655	.564
Conv.	.292	.691	.393	.293	.691	.396
Fiction	.400	.643	.490	.414	.621	.493
News	.513	.711	.592	.513	.709	.590
Av.	.434	.675	.510	.438	.669	.511

Table 6: Performance of a model trained with optimally-weighted examples with and without DC (difference in concreteness) features.

weighting regime. Here the effect of the difference in concreteness features is somewhat more pronounced for the essay data, with an average F1-score increase of 0.5 points, due to a 1.1 point average increase in precision along with 0.6-point drop in recall. Since auto-weighting generally leads to high-recall low-precision performance, improvement in precision is helping the system to achieve a more balanced performance.

The effect of the difference in concreteness features on the performance in the optimized weighting regime (Table 6) is less consistent across datasets; while we observe an improvement in precision in VUA data, the precision has dropped in the essay data, and vice versa with recall.

8 Results

In this section, we put together the different elements addressed in this paper, namely, the weighting regime, the different representation given to the concreteness feature relative to baseline, and the newly introduced difference in concreteness features. We compare performance to the baseline feature set (UPT+CBins) containing unigrams, POS features, topic features, and binned concreteness features (without thresholding), in an unweighted training regime, corresponding to the best feature set in Beigman Klebanov et al. (2014). These results are compared to the current best feature set (UPT+CUpDown+DCUpDown), in the optimized weighted training regime. The results are summarized in Table 7.

The overall effect of the proposed improvements is an absolute increase of 5.2 F1 points (9% relative increase) on essay data, on average, and 9.8 F1 points (24% relative increase) on VU Amsterdam data, on average.

9 Discussion

While the proposed improvements are effective overall, as shown in section 8 (Results), it is clear that the main driver of the improvement is the reweighting of examples, while the contribution of the other changes is very small (observe the small difference between the second column in Table 7 and the OptiWeight column in Table 3). The small improvement is perhaps not surprising, since the baseline model itself already contains a version of the concreteness features. Given the relevant literature that has put forward concreteness and difference in concreteness as important predictors of metaphoricity (Dunn, 2014; Tsvetkov et al., 2014; Gandy et al., 2013; Assaf et al., 2013; Turney et al., 2011), it is instructive to evaluate the overall contribution of the concreteness features over the UPT baseline (no concreteness features), across the different weighting regimes. Table 9 provides this information. The improvement afforded by the concreteness and

Data	UPT+ CBins unweighted (Baseline)			UPT+ CUpDown+ DCUpDown opti-weighted		
	P	R	F	P	R	F
A-B	.713	.351	.470	.593	.556	.574
B-A	.567	.491	.527	.485	.635	.550
Set A	.701	.478	.566	.561	.661	.604
Set B	.760	.592	.665	.662	.722	.690
Av.	.685	.478	.557	.575	.644	.605
Acad.	.631	.351	.419	.531	.655	.564
Conv.	.503	.241	.317	.293	.691	.396
Fiction	.551	.291	.378	.414	.621	.493
News	.640	.464	.536	.513	.709	.590
Av.	.581	.337	.413	.438	.669	.511

Table 7: Performance the baseline model UPT+CBins in the baseline configuration (unweighted) the UPT+CUpDown+DCUpDown model in opti-weighted configuration.

Expression	Conc. Adj	Conc. N	P(meta)
full educational [experience]	3.6	1.8	0.72
reach FULL [potential]	3.6	1.9	0.60
to its FULL [potential]	3.6	1.9	0.60
FULL [understanding]	3.6	1.9	0.60
FULL [truth]	3.6	2.0	0.60

Table 8: Instances of the adjective *full* in Set B that are predicted to be non-metaphors by the UPT model trained on Set A in the unweighted regime, while the UPT+CUpDown+DCUpDown model classifies these as metaphors. The noun that is recognized as being in the amod relation with *full* is shown in square brackets. FULL (small caps) indicates an instance that is annotated as a metaphor; lowercase version corresponds to a non-metaphor annotation.

difference-in-concrteness features is 1.4 F1 points, on average, for the unweighted and auto-weighted regimes for essay data and 0.6 F1 points, on average, for the VUA data; there is virtually no improvement in the optimized weighting regime.

To exemplify the workings of the concreteness and difference-in-concreteness features, Table 8 shows the instances of the adjective *full* observed in Set B where UPT predicts non-metaphor (P(metaphor)=0.41), while the UPT+CUpDown+DCUpDown model predicts metaphoricity (P(metaphor) > 0.5). We use logistic regression models trained on Set A data to output the probabilities for class 1 (metaphor) for these instances. The metaphoricity prediction in these cases is mostly correct; the one instance where the prediction is incorrect seems to be due to noise in the human annotations: The instance where the system is most confident in assigning class 1 label – *full* in " full educational experience" – has the adjective *full* labeled as a non-metaphor, which appears to be an annotator error.

In light of the findings in the literature regarding the effectiveness of concreteness and of difference in concreteness for predicting metaphoricity, it is perhaps surprising that the effect of these features is rather modest.

The incompleteness of the coverage of the concreteness database is one possible reason; 22% of instances in Set A do not have a concreteness value in the Brysbaert et al. (2013) database. Another possibility is that much of the information contained in concreteness features pertains to commonly used adjectives and verbs, which are covered by the unigram features. Mistakes made by the dependency parser in identifying eligible constructions could also impair effectiveness.

It is also possible that the concreteness ratings for adjectives in Brysbaert et al. (2013) data are somewhat problematic. In particular, we noticed that some adjectives that would seem quite concrete to us are given a concreteness rating that is not very high. For example, *round, white, soft, cold, rough, thin, dry, black, blue, hard, high, gray, heavy, deep, tall, ugly, small, strong, tiny, wide* all have a concreteness rating below 4 on a scale of 1 to 5. At the same time, they all have a fairly high value for the standard deviation (1.2-1.7) across about 30 responses collected per word. This suggests that when thinking about the concreteness of a word out of context, people might have conjured different senses, including metaphorical ones, and the judgment of concreteness in many of these cases might have been influenced by the metaphorical use. For example, if a person considered a concept like "dark thoughts" when assigning a concreteness value to *dark*, the

concept is quite abstract, so perhaps the word *dark* is given a relatively abstract rating. This is, of course, circular, because the perceived abstractness of "dark thoughts" came about precisely because a concrete term *dark* is accommodated, metaphorically, into an abstract domain of thinking.

Another possibility is that it is not concreteness but some other property of adjectives that is relevant for metaphoricity. According to Hill and Korhonen (2014), the property of interest for adjectives is subjectivity, rather than concreteness. A feature capturing subjectivity of an adjective is a possible avenue for future work. In addition, they provide evidence that a potentially better way to quantify the concreteness of an adjective is to use mean concreteness of the nouns it modifies – as if concreteness for adjectives were a reflected property, based on its companion nouns. A large discrepancy between thusly calculated concreteness and the concreteness of the actual noun corresponds to non-literal meanings, especially for cases where the predicted concreteness of the adjective is high while the concreteness of the actual noun is low.

10 Related Work

The field of automated identification of metaphor has grown dramatically over the last few years, and there exists a plurality of approaches to the task. Shutova and Sun (2013) and Shutova et al. (2013) explored unsupervised clustering-based approaches. Features used in supervised learning approaches include selectional preferences violation, outlier detection, semantic analysis using topical signatures and ontologies, as well as n-gram features, among others (Tsvetkov et al., 2014; Schulder and Hovy, 2014; Beigman Klebanov et al., 2014; Mohler et al., 2013; Dunn, 2013; Tsvetkov et al., 2013; Hovy et al., 2013; Strzalkowski et al., 2013; Bethard et al., 2009; Pasanek and Sculley, 2008).

A number of previous studies used features capturing concreteness of concepts and difference in concreteness between concepts standing in AdjN and VN dependency relations. The approach proposed by Turney et al. (2011) derives concreteness information using a small seed set of concrete and abstract terms and a corpus-based method for inferring the values for the remaining words. This information was used to build a feature for detection of metaphorical AdjN phrases; the methodology was extended in Assaf et al. (2013) and again in Neuman et al. (2013) to provide more sophisticated methods of measuring concreteness and using this information for classifying AdjN and VN pairs. Gandy et al. (2013) extended Turney et al. (2011) algorithm to be more sensitive to the fact that a certain concrete facet might be more or less salient for the given concept. Tsvetkov et al. (2014) used a supervised learning approach to predict concreteness ratings for terms by extending the MRC concreteness ratings. Hill and Korhonen (2014) used Brysbaert et al. (2013) data to obtain values for the concreteness of nouns, and derived the values for adjectives using average concreteness of nouns occurring with the adjectives in a background corpus. Apart from the exact source of the concreteness values, our work differs from these studies in that we evaluate the impact of the concreteness-related measures on an overall word-level metaphor classification system that attempts to classify every content word in a running text. In contrast, the approaches above were evaluated using data specially constructed to evaluate the algorithms, that is, using isolated AdjN or VN pairs.

The problem of machine learning with class-imbalanced datasets has been extensively researched; see He and Garcia (2009) for a review. Yang et al. (2014) and Muller et al. (2014) specifically evaluated the AutoWeighting technique on two different linguistic classfication tasks against a resamping-based technique, and found the former to yield better performance.

11 Conclusion

In this paper, we presented a supervised machine learning system for word-level classification of all content words in a running text as being metaphorical or non-metaphorical. The system provides a substantial improvement upon a previously published baseline, using re-weighting of the training examples and using features derived from a concreteness database. We observe that while the first manipulation was very effective, the second was only slightly so. Possible reasons for these observations are discussed.

Data	UPT			UPT+ CUpDown+ DCUpDown		
	P	R	F	P	R	F
A-B	.714	.337	.458	.712	.362	.480
B-A	.573	.480	.522	.565	.494	.527
Set A	.693	.462	.552	.699	.475	.564
Set B	.767	.576	.657	.760	.604	.672
Av.	.687	.464	.547	.684	.484	.561
Acad.	.635	.347	.418	.636	.356	.425
Conv.	.506	.240	.316	.487	.236	.309
Fiction	.549	.288	.374	.559	.309	.395
News	.641	.457	.531	.636	.466	.536
Av.	.583	.333	.410	.580	.342	.416
A-B	.513	.693	.590	.528	.713	.607
B-A	.400	.647	.494	.415	.670	.513
Set A	.498	.741	.594	.500	.747	.597
Set B	.568	.775	.655	.592	.773	.669
Av.	.495	.714	.583	.509	.726	.597
Acad.	.524	.651	.558	.525	.657	.562
Conv.	.292	.688	.392	.293	.691	.396
Fiction	.400	.600	.476	.411	.607	.486
News	.529	.665	.587	.530	.673	.590
Av.	.436	.651	.503	.440	.657	.509
A-B	.578	.597	.587	.593	.556	.574
B-A	.502	.612	.552	.485	.635	.550
Set A	.558	.659	.602	.561	.661	.604
Set B	.645	.705	.671	.662	.722	.690
Av.	.571	.643	.603	.575	.644	.605
Acad.	.521	.671	.565	.531	.655	.564
Conv.	.321	.614	.404	.293	.691	.396
Fiction	.398	.620	.481	.414	.621	.493
News	.506	.711	.586	.513	.709	.590
Av.	.437	.654	.509	.438	.669	.511

Table 9: Performance of a model without any concreteness features (UPT) and the model UPT+CUpDown+DCUpDown, in no-reweighting regime (top), auto-weighting (middle), and optimal weighting (bottom).

References

Dan Assaf, Yair Neuman, Yohai Cohen, Shlomo Argamon, Newton Howard, Mark Last, Ophir Frieder, and Moshe Koppel. 2013. Why "dark thoughts" aren't really dark: A novel algorithm for metaphor identification. In *Proc. IEEE Symposium Series on Computational Intelligence 2013*, Singapore.

Beata Beigman Klebanov and Michael Flor. 2013. Argumentation-relevant metaphors in test-taker essays. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 11–20, Atlanta, Georgia, June. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

Steven Bethard, Vicky Tzuyin Lai, and James Martin. 2009. Topic model analysis of metaphor frequency for psycholinguistic stimuli. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC '09, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Blanchard, Michael Heilman, and Nitin Madnani. 2013. SciKit-Learn Laboratory. GitHub repository, https://github.com/EducationalTestingService/skll.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2013. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, pages 1–8.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC*, pages 449–454, Genoa, Italy, May.

Jonathan Dunn. 2013. What metaphor identification systems can tell us about metaphor-in-language. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 1–10, Atlanta, Georgia, June. Association for Computational Linguistics.

Jonathan Dunn. 2014. Multi-dimensional abstractness in cross-domain mappings. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 27–32, Baltimore, MD, June. Association for Computational Linguistics.

Lisa Gandy, Nadji Allan, Mark Atallah, Ophir Frieder, Newton Howard, Sergey Kanareykin, Moshe Koppel, Mark Last, Yair Neuman, and Shlomo Argamon. 2013. Automatic identification of conceptual metaphors with limited knowledge.

Haibo He and Edwardo Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):12631284.

Felix Hill and Anna Korhonen. 2014. Concreteness and subjectivity as dimensions of lexical meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 725–731.

Dirk Hovy, Shashank Shrivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, and Eduard Hovy. 2013. Identifying metaphorical word use with tree kernels. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57, Atlanta, Georgia, June. Association for Computational Linguistics.

Hyeju Jang, Mario Piergallini, Miaomiao Wen, and Carolyn Rose. 2014. Conversational metaphors in use: Exploring the contrast between technical and everyday notions of metaphor. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 1–10, Baltimore, MD, June. Association for Computational Linguistics.

Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 27–35, Atlanta, Georgia, June. Association for Computational Linguistics.

Philippe Muller, Cécile Fabre, and Clémentine Adam. 2014. Predicting the relevance of distributional semantic similarity with contextual information. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 479–488, Baltimore, Maryland, June. Association for Computational Linguistics.

Yair Neuman, Dan Assaf, Yohai Cohen, Mark Last, Shlomo Argamon, Newton Howard, and Ophir Frieder. 2013. Metaphor identification in large texts corpora. *PLoS ONE*, 8(4), 04.

Bradley Pasanek and D. Sculley. 2008. Mining millions of metaphors. *Literary and Linguistic Computing*, 23(3):345–360.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. LDC Catalog No: LDC2008T19.

Marc Schulder and Eduard Hovy. 2014. Metaphor detection through term relevance. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 18–26, Baltimore, MD, June. Association for Computational Linguistics.

Ekaterina Shutova and Lin Sun. 2013. Unsupervised metaphor identification using hierarchical graph factorization clustering. In *Proceedings of HLT-NAACL*, pages 978–988.

Ekaterina Shutova, Simone Teufel, and Anna Korhonen. 2013. Statistical metaphor processing. *Computational Linguistics*, 39(1).

Gerard Steen, Aletta Dorst, Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification*. Amsterdam: John Benjamins.

Tomek Strzalkowski, George Aaron Broadwell, Sarah Taylor, Laurie Feldman, Samira Shaikh, Ting Liu, Boris Yamrom, Kit Cho, Umit Boz, Ignacio Cases, and Kyle Elliot. 2013. Robust extraction of metaphor from novel data. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 67–76, Atlanta, Georgia, June. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51, Atlanta, Georgia, June. Association for Computational Linguistics.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland, June. Association for Computational Linguistics.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xuesong Yang, Anastassia Loukina, and Keelan Evanini. 2014. Machine learning approaches to improving pronunciation error detection on an imbalanced corpus. In *Proceedings of IEEE 2014 Spoken Language Technology Workshop, South Lake Tahoe, USA*, pages 300–305.