Comparison of the Three Algorithms for Concreteness Rating Estimation of English Words

Vladimir V. Bochkarev, Stanislav V. Khristoforov, Anna V. Shevlyakova, Valery D. Solovyev

Kazan Federal University, Kremlyovstaya 18, 420008 Kazan, Russia,
vladimir.bochkarev@kpfu.ru, stanislav.khristoforov@tgtdiagnostics.com,
AVShevlyakova@kpfu.ru, Valery.Solovyev@kpfu.ru

Abstract: The paper compares three algorithms for concreteness rating estimation of English words. To train and test the models, we used a number of freely available dictionaries containing concreteness ratings. A feedforward neural network is employed as a regression model. Pre–trained fastText vectors, data on co–occurrence of target words with the most frequent ones, and data on co–occurrence of target words with functional words are used as input data by the considered algorithms. One of the three algorithms was proposed for the first time in this article. We provide detailed explanations of which combinations with functional words are the most informative in terms of concreteness ratings estimation for English words. Although the rest two algorithms have already been used for estimation of concreteness ratings, we consider possible ways to update them and improve the results obtained by a neural network. Thuswise, we use stochastic Spearman's correlation coefficient as a criterion for stopping of training. All three algorithms provided good results. The best value of Spearman's correlation coefficient between the value of the concreteness rating and its estimate was 0.906, which exceeds the values achieved in previous works.

Keywords: concreteness rating; abstractness; neural networks; fastText; word co–occurrence; English

1 Introduction

The issue of word concreteness has been the focus of attention of many academic disciplines for several decades. It is extensively studied in linguistics, psychology, psycholinguistics, medicine, neurophysiology, philosophy, and pedagogy [1]. The way abstract and concrete concepts are represented is a fundamental problem that has been debated in psychology, psycholinguistics, and neuro–physiology. A comprehensive review article [2] notes that the problem of representing abstract concepts is a crucial challenge for any theory of cognition.

Dictionaries with concreteness ratings of words are used to investigate this problem. For example, there are two large dictionaries of the English and Dutch languages created on the basis of native speakers‘ responses. Each of them includes approximately 40 thousand words [3, 4]. The dictionaries for other languages are tens of times smaller, specifically the Russian dictionary contains only 1000 words [5].

There are different approaches to the definition of abstractness and concreteness. It can be defined as (1) general, generic, not specific, and (2) lacking sense experience [6]. According to [6], nouns are considered concrete if they denote people, places and things and refer to a perceptible entity. If they cannot be experienced by our senses, they refer to more abstract concepts. Similar view on abstractness/concreteness is presented in [7], that is, abstract nouns are those that do not have denotata in real physical world and cannot be percepted. Criteria and norms that allow one to refer a word to an abstract or concrete concept are of great value for cognitive science.

However, instructing and asking participants of the experiments to validate the developed norms is a time– and labour–consuming process. Though there were some advances in this field such as Mechanical Turk, a service that allows collecting and processing data, as well as predicting values of concreteness. Using experts' responses is still not an easy task. It is proved by the fact that the largest concreteness dictionaries provide ratings for a relatively small number of words. Creation of large text corpora and development of machine learning methods can contribute much to the solution of this problem.

One of the possible options is extrapolation of ratings obtained by expert assessment to a wider range of words. The degree of usefulness of such extrapolated ratings depends on how effectively the extrapolation procedure is realised. Extrapolated ratings are most useful when only small datasets of human judgments are available. Developing methods that allow for high–quality extrapolation from actual human judgments is a sort of breakthrough [8].

As stated above, it is too expensive and time–consuming to conduct experiments when experts determine concreteness ratings of large number of words (tens of thousands and more). Any corpus created using human ratings would be relatively small. However, large corpora are needed to solve practical tasks as the larger the corpus is, the more reliable are its data. The study objective is to develop a computational model for prediction of concreteness ratings. Using the model will be more efficient than conducting the experiments as it will allow one to obtain more trustworthy ratings in far lesser amount of time.

Three algorithms for estimating concreteness ratings of English words are compared in our work. These algorithms differ by the type of the used vector representation of words. Two of these algorithms have already been used to estimate concreteness ratings. The third algorithm that uses data on co–occurrence of the target words with the context ones is proposed for the first time in this article. Therefore, we briefly discuss the reasons that make the introduced algorithm effective.

2 Related Works

Most studies on extrapolation of expert estimates and automatic expansion of the dictionary have been performed for the English language. The main idea and the study stages are as follows:

(1) A set of words with expert ratings of the degree of concreteness/abstractness is selected; some of the words are used to train the extrapolation method and the rest ones are used for testing.

(2) Words are represented by vectors in some semantic space.

(3) Some extrapolation method is applied to the data.

(4) The estimates obtained on the test set of words are compared with the expert ones.

Dictionaries presented in [3, 9] are selected as a set of words with expert ratings for the English language. In early studies, LSA was chosen as the semantic space; in recent works, a skip–gram model has been used for such purposes. Different types of Regression Models (SVM, neural networks, etc.) are often used as methods of approximation. Spearman‘s or Pearson’s correlation coefficient between ratings of concreteness and their estimates is utilized as accuracy measure of the model in most works. Table 1 summarizes the research results for the English language. The table includes papers in which correlation coefficients of 0.7 and higher were obtained.

Comments to the table:

(1) Some papers presented in Table 1 provide comparison of various methods. In this case, the best method is put in bold.

(2) Some papers use a very small set as a teaching one (the core). In this case, the number of concrete and abstract words are shown in parentheses in the “volume” column.

(3) The employed type of correlation coefficient is shown in the last column. Some papers use a binary classification (concrete/abstract words) instead of ranking. In this case, the value of the accuracy parameter is calculated instead of Spearman’s or Pearson’s correlation coefficient.

(4) The neural network was trained on sentences, not on separate words in [10] (the 7th row in the table). They used 800,000 sentences that contain 2580 words rated as abstract or concrete.

Table 1

Related works and obtained results for English

Paper, year	Corpus	Semantic space	method	volume train/test	correlation
[11], 2011	[9], English	LSA	A step-wise regression analysis	3,521 67%/33%	0.802 (Pearson)
[12], 2011	[9], English	LSA	Cosine similarity	4,295 0.9% (20-20) / 50%	0.822 (Spearman)
[13], 2013	[9], English	vector space representations from [14]	logistic regression classifiers	2,450 98%/2%	0.76 (accuracy)
[15], 2015	[3], English	LSA, topic model, a hyperspace analogue to language (HAL)-like model, a skip-gram model.	**k-nearest neighbours,** random forest	37,058 25%/75%	0.796 (Pearson)
[8], 2017	[3], English	a skip-gram model	step-wise regression model	37,058 50%/50%	0.829 (Pearson)
[16], 2018	[3], English	a skip-gram model	algorithm SentProp [17]	14,329 0.2%(15-15)/99.8%	0.70 (Spearman)
[10], 2018	English Wikipedia[1]	GloVe	Naive Bayes, Nearest neighbor, **RNN**	2580 81%/19%	0.740 (Pearson)
[18], 2018	[3,9], English	fasttext	**SVM**, feedforward networks	22,797 67%/33%	0.887 (Spearman)
[19], 2019	[3,20,21], English	fasttext	SVM	32,783 [3] / 2,005 [20,21]	0.902 (Pearson)

Now we note some results obtained in the above–mentioned works that were not included in the table. One of the first papers where high results were obtained is [22] (the Spearman's correlation coefficient is 0.64). The study [23] stands apart from the rest ones since it carries out extrapolation not within one language but between languages using a multilingual skip–gram model. In this case, the extrapolation method is trained on the full set of available data from one language. It is stated in [23] that the data were extrapolated on 77 languages; however, the

[1] English Wikipedia, May 2017 dump.

data on all languages are not presented. When extrapolating estimates from English to Dutch, Pearson's correlation coefficient with the expert estimates in Dutch from [4] equaled 0.76. One of the observations described in [23] is that more frequent nouns and verbs are less concrete in both English and Dutch.

Besides English and Dutch, both experts ratings and automatically generated ones were used to create dictionaries for other languages. For example, automatically obtained ratings for Chinese, Persian and Russian are presented in [24, 25] and [5, 26], respectively. If the algorithm starts with a small core (see papers [16, 12]), the question arises about selecting words for the core. The core of a fixed size in [16] included most frequent and most concrete and abstract (according to the expert ratings) words. The core of a fixed size in [12] contains 40 words. It is formed iteratively starting from an empty set and sequentially adding words that are in best correlation with abstract and concrete words from the training set. It is shown that if the core is expanded to 100 words, the Pearson's correlation coefficient on the test set will decrease.

3 Data and Method

The BWK base [3] was used as a source of ratings. This base provides concreteness ratings for about 40,000 words and word combinations. To test the trained models, we also utilized concreteness ratings from the MRC database [9], the Toronto Word Pool datasets [20] and the base created by Paivio, Yuille and Madigan [21] that provide concreteness ratings for 4239, 1093 and 925 words, respectively (the bases will be further abbreviated as MRC, TWP and PYM).

We use vector representations of words that have been developed within the framework of distributive semantics. The distributive semantics approach assumes that there is a correlation between distributional similarity and meaning similarity [27, 28, 29]. Currently, the most widely used methods are based on vector models of neural networks [30]. However, simpler representations based on explicit word vectors are also applied (see overviews [31]).

The first of the compared methods of word representations uses word embedding algorithms. Good reviews of word embedding methods can be found, for example, in [31, 32]. One of the main results in this area was described in the article [30] that introduced the word2vec model. Employing stochastic algorithms for learning artificial neural networks, the authors managed to obtain low–dimensional (with a dimension of 250-300) vector representations of words, which also implemented various semantic relations between them. One more significant achievement in this area was the fastText algorithm proposed in [33]. Combined usage of the word2vec model and subword information significantly reduces the time of model training and provides better result. The authors of [34] have granted free access to four types of sets of pre–trained vectors (the dimension of word representation is 300). The sets differ by the source on which they were trained (Wikipedia 2017, UMBC webbase corpus or Common Crawl) and by whether subword information was used or not. Following the recommendation given in [19], we use vectors that do not include subword information.

Pre–trained fastText vectors have already been used in many studies on estimation of concreteness ratings. For example, two recent studies [18, 19], in which the highest results in the accuracy of concreteness rating estimation were obtained (see Table 1), use the fastText vectors as input data.

The second of the compared algorithms employs explicit word vectors. We use vector representations based on co–occurrence with the most frequent words (CFW). The CFW method is described, for example, in [35, 36]. The CFW method was applied in [37] to estimate the concreteness ratings of Russian words. In accordance with this approach, the target word is represented by a frequency vector of bigrams that include the target word and one of the context words. The CFW method uses a given number of the most frequent words as context words.

In our work, we use unigram and bigram frequency data extracted from the Google Books Ngram corpus [38]. The English (Common) subcorpus of Google Books Ngram includes texts of 16.6 million books published between 1470-2019 that contain approximately 2 trillion words. Currently, it is the largest corpus of the English language. Since we use the GBN corpus data, to make the list of context words, we selected 20,000 words that were most frequently used in GBN between 1900-2019.

Frequencies of combinations of each target word with each context word were extracted from the corpus (if some word combination was absent from the corpus, the corresponding frequency was considered equal to 0) As two types of bigrams are possible (with the target word in the first (Wx) and second places (xW)), we obtain two vectors with a dimension of 20,000. The last step is concatenation of these two vectors. Thus, a vector with a dimension of 40,000 is obtained for each target word. Besides ordinary bigrams, which are pairs of consecutive words, GBN contains information on syntactic bigrams [39]. We compared types of vector representation obtained using data for both ordinary and syntactic bigrams.

As a rule, the resulting vectors are very sparse (contain a large number of zeros); however, they carry all information about the co–occurrence of the target word with the most frequent ones. The drawback of this method is high dimension of the resulting vector representation, which can cause significant problems in the process of training neural network models (especially for fully connected networks) and lead to overfitting. Therefore, if this type of word representation is used, it is important to ensure good regularization of the model during the training process.

The third of the compared algorithms is proposed in this article for the first time. We also use explicit word vectors. The scheme of the proposed algorithm is analogous to the CFW method with the only difference that we use functional words (not always the most frequent ones) as context words. The proposed algorithm will further be called the CSW algorithm (co–occurrence with stop–words). It was abbreviated as CSW for the following reason. If we abbreviated it as CFW (co–occurrence with functional words), this could lead to misunderstanding as the abbreviation CFW already exists and is mentioned in this paper. Therefore, we replaced the letter F by S, where S refers to stop–words. By stop–words we understand functional words presented in [40]. We borrowed the list of 307 functional words from [40]. It includes articles, conjunctions, particles, prepositions, as well as numerals, auxiliary verbs, some adjectives, pronouns, etc.

We used the GBN corpus to extract frequencies of bigrams that include the target words and one of the functional words. Thus, we obtained a vector representation of dimension 614 for each target word (taking into account bigrams of the type Wx and xW).

The pre–trained fastText vectors can be directly fed into the neural network input; however, when using explicit word vectors, appropriate preprocessing is required. The first problem is a large range of change in bigram frequencies. For example, a set of vectors that we used contains frequency values from 40 to $1.8 \cdot 10^{10}$. The second problem results from the fact that values of absolute frequencies depend on a corpus size; and if it is required to use the obtained model on other data, the vectors need to be normalized. Based on the experience of previous works (see, for example, [41]), two preprocessing methods were chosen.

The first one proposed in [42] assumes that frequency values are used to calculate the corresponding Pointwise Mutual Information values.

$$PMI_{i,j} = \log_2 \frac{f_{i,j}}{f_i f_j} \tag{1}$$

Here f_i is the relative frequency of the i-th target word, f_j is the relative frequency of the j-th context word, f_{ij} is the relative frequency of the bigram in-cluding the i-th target word and the j-th context word. On the one hand, PMI is composed of relative values and does not depend on the size of the employed corpus; on the other hand, it provides compactification of the dynamic range due to the presence of a logarithm.

The second considered preprocessing method is taking a simple logarithm of frequency vectors. This technique also allows one to reduce the dynamic range of the vector input values; however, it does not eliminate the dependence on the size of the em-ployed corpus. Nevertheless, this preprocessing method has shown good results in several tasks [41]. To perform preprocessing correctly when frequency value equals zero, 1 is added to the frequencies before taking the logarithm:

$$\log_2\left(F_i + 1\right) \tag{2}$$

where F_i is the frequency of the bigram at the *i*-th position of the input vector.

The traditional fully connected feedforward network [43] was chosen as a model that solves the problem of estimating concreteness ratings of words. It consisted of 4 hidden layers; each of them contained 128 neurons. Each neuron in the hidden layer used ELU [44] as activation function. The output layer contained 1 neuron with an identically linear activation function.

Despite the fact that we used the same neural network architecture in all three cases, the number of weights in the network is significantly different in each case. The number of variable network parameters was about 5.1 million for the CFW method. Due to the large dimension of the input vector (D = 40,000), almost 99.5% of all free parameters of the model were concentrated in the input layer. Therefore, much attention was paid to regularization when training this model. Regularization was also used for the models based on the rest two methods (the dimension of the input vectors equaled 300 (D = 300) and 614 (D = 614), respectively); however, its impact on the training process was significantly lower.

The main regularizer was the dropout layer placed between the input and the first hidden network layer. Stochastic disabling of connections between neurons provides regularization and prevents overfitting of the neural network [45]. The regularization parameter of 0.3 was chosen for the model based on the CFW method. Thus, only random 70% of all connections of the layer were used and corrected at each training iteration. The dropout parameter was significantly lower (0.1) for the rest two methods. Beyond that, L1–regularization of all hidden layers was additionally employed when the CFW method was used. This allowed us to obtain a sparser representation as well as to reduce the tendency of this model to overfit.

The mean square error (MSE) between the target value of the concreteness rating and the resulting network estimate was chosen as a loss function for all three types of models. The model was trained based on stochastic gradient descent by the Adam [46] method. At each training iteration, random 128 examples from the training sample formed a training batch, the root mean square error of which was minimized by the network. Simultaneously, a similar batch of the same size was generated from the test sample for network validation. When the target loss did not decrease by more than 10% during 1,000 iterations of updating the weights, we artificially reduced the learning rate parameter [47]. Each time this condition was met, the learning rate was reduced by half. This allowed improving the network results obtained at the last stages of its training when the values of the target loss function have practically not changed.

In addition to the loss function, Spearman's correlation coefficient was chosen as an additional metric and calculated on the test sample during the training process.

Spearman's correlation coefficient on the entire test sample is calculated significantly longer in comparison with the time spent on the forward and backward signal propagation through the network. Since this metric had to be read after each iteration of the network weight adjustment, its stochastic version was implemented.

Spearman's correlation coefficient was calculated using random 2048 samples from the test sample. At that, such a truncated metric turned out to be a representative estimate of the Spearman's correlation coefficient calculated for the entire test sample. This metric was used to control the overfitting of the network, as well as a criterion for stopping the training process. After the values of stochastic Spearman's correlation coefficient reached a plateau, the training stopped after 1000 updates of the weights. The network weights corresponding to the highest value of this metric were further used to test the training results.

The PyTorch [48] automatic differentiation library was used as a framework for training the neural network model.

Thus, we trained and tested 10 models in total. They were represented by two models for the case of using fastText vectors trained on Wikipedia and CommonCrawl, four models for each case of employing the CFW and CSW methods (for ordinary and syntactic bigrams, and two types of input data preprocessing).

As it was stated above, in many papers, the obtained accuracy is estimated by calculating Spearman's or Pearson's correlation coefficients between a concreteness rating and its estimate. Some papers (see, for example, [19]) use Kendall's correlation coefficient. To simplify comparison with prior works, in most cases, we provide values of the three coefficients. In our opinion, the use of Spearman's correlation coefficient is the most justified in this case since its employment is not associated with certain assumptions about distribution of the analyzed data [49].

4 Results

As mentioned in the previous section, 20% of words presented in the [3] database were selected for testing. These words were not used for training. The selection of words for the test sample was carried out randomly; and the words included in it have the same frequency distribution and part–of–speech distribution as the words in the training sample. The test sample included 7406 words for the models that use bigram frequencies extracted from GBN as input data and 6815 words for the models that employ pre–trained fastText vectors. Here, 7406 words are 20% of 37,030 words that are found both in the BWK base and GBN; and 6815 words are 20% of 34,076 words that are found in the BWK base and have fastText vectors.

As an example, Figure 1 shows bidimensional distribution of concreteness rating values and their CFW (ordinary bigrams, PMI) estimates. The figure illustrates that the quality of rating estimation is quite high.

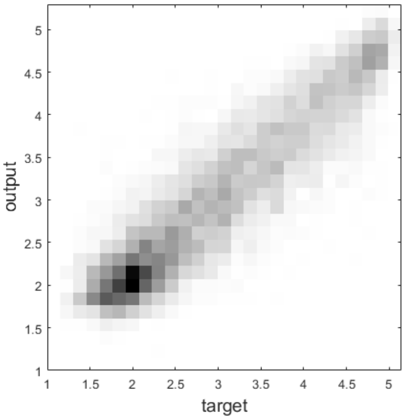

Figure 1
Bidimensional distribution of concreteness rating values and their estimates obtained using CFW (ordinary bigrams, PMI)

The values of the correlation coefficients between the values of the concreteness rating and its estimation were calculated on the test set for each of the 10 models. The results obtained using the CFW and CSW methods are shown in Table 2.

Table 2
Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values and their estimates obtained using the CFW and CSW methods (for the BWK dataset)

Set of Vectors		PMI	$\text{Log}_2(1+x)$
CFW, ordinary bigrams	r	0.899	0.901
	ρ	0.884	0.884
	τ	0.702	0.701
CFW, syntactic bigrams	r	0.902	0.902
	ρ	0.888	0.887
	τ	0.706	0.704
CSW, ordinary bigrams	r	0.890	0.883
	ρ	0.875	0.868
	τ	0.688	0.680
CSW, syntactic bigrams	r	0.890	0.878
	ρ	0.873	0.861
	τ	0.686	0.671

It is obvious that the CFW method has a slight advantage over CSW. It should be noted that the results obtained using ordinary and syntactic bigrams almost do not differ. The two data preprocessing methods also provided approximately the same results when we used the CFW method. When the CSW method was employed, the use of PMI vectors significantly improved the accuracy. These results differ form ones obtained in [41] and shows that solving different tasks may require different preprocessing methods.

Table 3 shows the results of testing the models based on fastText vectors. Better results are obtained when using vectors trained on the CommonCrawl corpus.

Table 3

Pearson’s (r), Spearman’s (ρ) and Kendall’s (τ) correlation coefficients between the concreteness rating values and their estimates for the [3] dataset using fastText vectors

Set of Vectors	r	ρ	τ
CommonCrawl	0.916	0.906	0.729
Wikipedia	0.901	0.893	0.710

For each of the three methods, we selected the variant that provided best results. For the CFW method, it we used syntactic bigrams and PMI (=0.888); for the CSW method, we employed ordinary bigrams and PMI (=0.875); and for the method employing fastText vectors, we utilized vectors pre–trained on the CommonCrawl corpus (=0.906). For each of the described cases, the neural model was trained 10 times and tested. Standard deviation of Spearman‘s correlation coefficient between the concreteness rating and its estimate was $4\cdot10^{-3}$ for the CFW method, $3.5\cdot10^{-3}$ for the CSW method, and $1.7\cdot10^{-3}$ for the fastText method. Comparing the obtained values with those shown in Tables 2,3, one can see that the differences in accuracy between the three methods are not large, however, they are statistically significant.

Ideally, comparative testing of different methods should be carried out using corpora of the same size. Unfortunately, in practice, one has to use available tools and datasets. In our case, the size of the CommonCrawl corpus that was used to obtain fastText vector representation is about 3 times smaller than that of the English (common) subcorpus of GBN employed to obtain vectror representations by the CFW and CSW methods. However, this does not cause difficulties in determining which of the compared methods showed the highest accuracy. The highest result was obtained using the pre–trained fastText vectors. If we used a corpus which 3 times exceeds the size of CommonCrawl, we would probably expect even more increase in accuracy.

Now we consider how estimation accuracy of concreteness rating depends on word frequency. To perform a comparative analysis of the three algorithms, we select the variant that provides the highest values. These variants are the use of syntactic bigrams and PMI (for the CFW method) and the use of fastText vectors obtained on the CommonCrawl corpus (for the CSW method).

The following approach was used to analyze the dependence of accuracy on frequency. We sort the words in the test sample in descending order of frequency. After that, we calculate Spearman‘s correlation coefficient between the rating values and its estimates in a sliding window with a length of 1000. Each position of the window defines a certain frequency range. We take the geometric mean of frequency of words that fell into the sliding window for visualization. The obtained results are shown in Figure 2.

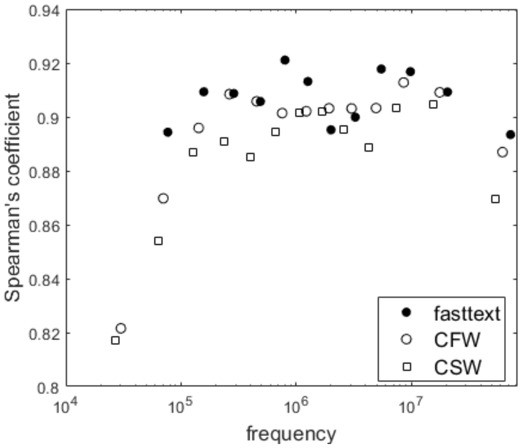

Figure 2
Dependence of Spearman‘s correlation coefficient between the values of concreteness rating and their estimates on word frequency

The figure shows that the frequency dependence is weak in a wide frequency range (from 105 and higher). At that, some advantage of estimates obtained using fastText vectors is observed. The accuracy of the estimation obtained using the CFW and CSW methods starts decreasing with frequency decrease. It is a complicated task to analyze the accuracy of the fastText method for this frequency range since there are few rare words in the corresponding test sample.

Table 4
Spearman’s correlation coefficients (ρ) between the values of the concreteness rating and their estimates for different parts of speech

Method	NOUN	ADJ	VERB
CFW	0.899	0.774	0.804
CSW	0.886	0.733	0.770
fastText	0.912	0.785	0.826
percentage of case, %	55	21	15

Table 4 presents data on the accuracy of concreteness rating estimation for each of the main parts of speech separately. For each of the three methods, we choose the variant mentioned above when we considered the impact of frequency on the estimation accuracy. The last line in the table shows the percentage of words related to one or another part of speech in the test sample. The highest accuracy of concreteness rating estimation is achieved for nouns. The Accuracy for verbs and even more so for adjectives is lower. One of the reasons is that there are significantly more nouns in the training set than verbs and adjectives. Therefore, the model was better trained for nouns.

Let us compare the level of accuracy achieved by us with the results obtained in previous works. As far as we know, currently, the highest accuracy in estimating the specificity rating has been achieved by the authors of [18, 19]. Testing is carried out on the BWK data in [18]. This work employs fastText vectors as input data; and comparative testing of two regression algorithms is carried out using the SVM method and the feed–forward neural network. The obtained values of the Spearman's correlation coefficient between the values of the concreteness rating and its estimate for these algorithms are 0.887 and 0.879, respectively. It should be mentioned that the second of the algorithms described in [18] is similar to one of the methods that we compare in our work (it uses the fastText vectors); however, the level of accuracy we obtained is noticeably higher (0.906 versus 0.879). It can be assumed that this is primarily due to the use of a different criterion for stopping of training, as well as the difference in the employed regularization methods.

Table 5

Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values and their estimates for different datasets

Method		TWP	PYM	MRC
CFW	r	0.910	0.915	0.900
	ρ	0.875	0.926	0.901
	τ	0.698	0.764	0.722
CSW	r	0.909	0.916	0.891
	ρ	0.886	0.913	0.895
	τ	0.710	0.747	0.711
fastText	r	0.890	0.916	0.892
	ρ	0.878	0.920	0.896
	τ	0.696	0.745	0.712
fastText+SVM [19]	r	0.881	0.902	-
	ρ	-	-	-
	τ	0.698	0.741	-

The test results are given for the MRC database, as well as for TWP and PYM datasets in [19]. We select words from these three datasets, which are also present in the test sample. Table 5 shows the values of the correlation coefficients between the estimates obtained by the neural network and the concreteness ratings extracted from the TWP, PYM, and MRC. Similar to [19], we compare the rating values not with the target values extracted from BWK but with the values of the ratings in these three datasets.

The table shows that all three methods under consideration provide better results than those described in [19]. Charbonnier and Wartena [19] also raised an important question about the limit of the achievable accuracy in estimating the concreteness ratings. Therefore, the authors [19] analyzed the level of correlation between the rating values given in different datasets. Table 6 shows the values of the correlation coefficients between the concreteness rating values given in the BWK and the ratings of the same words presented in the MRC, TWP and PYM databases. One can see that we obtained the values of the correlation coefficients between the target value of the rating and its estimation (see Table 5) that almost reach the values shown in Table 6. In many cases, the difference is only a few thousandths.

Table 6

Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values given in BWK and other datasets

Dataset	r	ρ	τ
TWP	0.913	0.899	0.736
PYM	0.936	0.932	0.770
MRC	0.919	0.921	0.748

This seems surprising since the models were trained only on BWK ratings. To understand the reason, we select those words from the test sample that are also found in one of the other datasets (TWP, PYM or MRC). For example, the test sample contains 199 words that are also present the TWP dataset. In 114 cases of 199 (or in 57.3% of all cases), the estimate obtained by the neural network deviates from the target value in the same direction as the value of the concreteness rating in TWP (here we use estimates obtained using the CSW, PMI methods and ordinary bigrams). A priori, it is natural to assume that deviations of the estimation of the concreteness rating from the target value upward and downward are equally probable. If this hypothesis is correct, then the number of cases where the estimates obtained by the neural network lie closer to the values from the TWP than the target values (extracted from the [3] base) should obey the binomial distribution with the parameter 0.5. It is easy to calculate that the p–value for this case is 0.0234. The test sample contains 166 words that are also included in the PYM dataset. The estimates of concreteness rating of 93 words from 166 (56% of all cases) are closer to the PYM ratings than the target values. In this case, the p–value is 0.070. Finally, the MRC base contains 773 words that are also present in the test sample. In 463 cases (56.4% of all cases) the estimates deviate from the target values so that their difference from the target values given in MRC decreased. The p–value for this case is $2.08\cdot10^{-4}$. Thus, at any reasonable level of significance, the null hypothesis that the estimate is equally likely to deviate from the target value upward and downward should be rejected.

Thus, although the neural network was trained only on BWK data, due to the ability of the neural network to generalize, the rating estimates often deviate from the target ones in such a direction as to approach the rating values from other datasets. That is, the neural network seeks to "correct" errors occurred during rating estimation performed by individual research groups.

6 Interpretation of Results

It is a surprising fact that using CSW provides accuracy that is slightly lower than that obtained by the other two methods. Indeed, less information is fed to the input of the neural network in this case than employing the other two considered algorithms. From the utilized list of functional words, 299 (or 97.4%) are also included in the list of 20,000 most frequent words applied in the CFW method. Adding 19,700 more context words to the list of context words allows us to raise Spearman's correlation coefficient of the concreteness rating and its estimate from 0.875 to only 0.888. In this section, we will try to explain why it becomes possible to achieve high accuracy in estimating the concreteness rating using the CSW method.

We repeated the calculation of the ratings, disconnecting one of the inputs of the neural network in turn. This was done by feeding the corresponding input zero values for each target word. Then, we calculated the increments of Spearman's correlation coefficient by formula 3:

$$\Delta\rho^{(i)} = \rho^{(i)} - \rho \tag{3}$$

Here ρ is Spearman's correlation coefficient for a network using all inputs, and $\rho^{(i)}$ is Spearman's correlation coefficient for a network with the disabled i–th input. Notice that the more useful a type of bigram is for determining the concreteness rating, the more significant the drop in Spearman's correlation coefficient will be when the corresponding input is turned off.

At the next stage, we sorted all bigrams in the descending order of the $\Delta\rho^{(i)}$ increments. The words (more precisely, the construction with the words) that have the greatest influence on the concreteness ratings of the target words were at the top of the list. As the words referring to different parts of speech may have different "influential" context words, we performed the described calculations for each of the studied part of speech.

We analysed 614 contexts the studied words appear in and ranged the context words (Wx, xW types) according to their contribution to the concreteness ratings.

We formed 3 lists of words. The first list included ranged context words that influence the concreteness ratings of nouns, the rest ones consisted of words that influence the ratings of verbs, adverbs, and adjectives, respectively. The task is to describe the most typical group of words form the list without detailed semantic analysis, though we give some clues why the studied words are in the list taking certain place.

The first group we analysed was the list of combinations of functional words with nouns. The most "influential" word in this group is the indefinite article *a* ("a+X") that is usually used with concrete countable nouns in the considered construction. The definite article *the* takes the third place in the rating that can be used both with abstract and concrete nouns, however, we can say that nous preceded by *the* are often more concrete than those with the zero and indefinite articles.

The third and thirteenth top constructions are "X+of" and "of+X", respectively. They form the genitive construction that usually shows relations between two nouns, such as mereology, taxonomy, valency, etc. Used both with concrete and abstract nouns (*out of curiosity*), we may hypothesize that this construction is more typical of concrete nouns. The top construction "X+from" can be compared to the genitive construction considering mereology, it describes the part divided from the whole. It resembles extraction of something or somebody from something. We suppose that it is more often used with concrete nouns in the studied construction. The construction "X+with" often describes the whole with the added part. It may be used with concrete/abstract nouns and in set expressions (*in love with somebody),* however, concrete nouns are more expected to be used in this structure. The preposition on ("X+on") assumes something/somebody locating on something/somebody, i.e contact between the figure and the ground [50]. It is used both with concrete and abstract nouns (*shame on yo*u), however, concrete sense occurs more often [51]. Primarily function of all prepositions is to describe spatial relationships between concrete nouns though abstract uses are also common. The considered top list prepositions are *beside* ("X+beside"), *within* ("X+within"), *among* ("X+among"), *onto* ("X+onto"), *into* ("into+X"), *towards* ("towards+X") etc. Their contribution to abstract/concrete correlation is valuable.

Besides prepositions, the list of context words included adjectives like *each* ("each+X") and *every* ("every+X") denoting "every one of two or more considered individually or one by one", "being one of a group or series taken collectively" (https://www.dictionary.com). Such words are usually used with countable nouns that denote concrete nouns. Thus, they provide higher correlations in abstractness estimation.

Numerals also have contribution to the ratings. The structures like "two/three/four five+X" and "X+two/three/four/five" with the latter ranked higher in the list. The word first is at the top of the list.

Possessive pronouns and demonstrative adjectives ("my/their/our/your+X" and "this/these+X"), are also among fifty most "influential words". Possessive pronouns refer to something that we have or that relate to us (in a wide sense). It seems that we possess something visible and concrete though we can, for example, feel something and describe as “my feeling”. If considering theme/rheme relations, nouns determined by possessive pronouns are more concrete in the contexts than undetermined ones. Demonstrative adjectives refer to different type of objects, but the grammatical structure implies that *this* refers to one object and *these* – to several objects. *This* can combine with both concrete and abstract nouns depending on the context. However, these usually refer to several concrete objects.

The quantifiers *much* (“much+X”) and *many* (“many+X”) are in the middle of the list. *Much* is used with singular uncountable nouns that are often abstract nouns; *many* is used with plural nouns that are usually concrete. Therefore, they contribute much to concreteness ratings estimation.

There are conjunctions (“and+X”, “or+X”), reflexive pronouns (“ourselves+X”), auxiliaries (“X+will”). They are less “influential” considering concreteness ratings.

When analysing concreteness ratings of adjectives, we should bear in mind nouns because adjectives modify nouns. If the adjective occurs with more abstract noun than usual, it is an indicator of metaphoricity [52], therefore, its sense becomes more abstract. Thus, the construction “a+adjective” implies that there is some noun behind. If we consider nominal predicates, the modified noun is before the adjective. The ranged context words for adjectives showed some correlation with the ranged list created for nouns. For example, the articles (“a/the+X”), demonstrative adjectives (“this/these+X”), quantifiers (“X+many”) are also at the top of the list. However, detailed analysis lies in the linguistic domain.

We also ranked the context words that influence the correlation of verbs concreteness. Among the most “influential” ones are reflexive pronouns (“X+themselves, ourselves, herself’, “yourself+ X”), and adverbial modifiers (“X+again”, “already+X”).

Conclusions

We compared three algorithms for estimating concreteness ratings of English words. To train and test the models, we used a number of freely available databases that contain concreteness ratings [3, 9, 20, 21].

Spearman’s correlation coefficient between the concreteness rating and its estimate of 0.906 was obtained on the test sample of words included in the BWK database [3]. Even higher correlation values were obtained for high–frequency words included in the [21] dataset and the MRC Psycholinguistic Database. The achieved level of accuracy exceeds the values obtained in previous works.

Increase in accuracy became possible due to some improvements of the training process of the models. The most significant improvement was the use of stochastic

Spearman's correlation coefficient that was employed as the second metric in the training process. The value of this metric was used as a criterion for stopping the training process, as well as for choosing the best iteration.

The comparison of the three tested algorithms shows that each of them has its own advantages. The algorithm that uses fastText as input data showed the highest accuracy and can be used to extrapolate concreteness ratings to a wide range of words based on synchronous data. The other two algorithms showed slightly lower accuracy. However, their advantage is easy adaptation to diachronic data, for example, when using large amount of data from the Google Books Ngram corpus. A recent work [16] showed that the values of the concreteness rating of some words change significantly over time. This phenomenon is of great interest and needs further study. The CSW method seems especially promising for diachronic studies since combinations with functional words are usually quite frequent. Besides possible practical applications of the CFW method, the fact that its accuracy is practically not inferior to the accuracy of the other two considered methods is of interest from the point of view of theory.

Another advantage of the algorithms using explicit word vectors is the ease of interpretation of the obtained results. If a model employing word embeddings is a black box for models based on explicit word vectors, we can determine which combinations occurring in the corpus increase or decrease the concreteness estimates.

The results obtained in this work can be used to create large dictionaries with concreteness ratings of words and other semantic and psychological variables, which is important for many practical applications.

Acknowledgement

This research was financially supported by Russian Foundation for Basic Research (grant 19–0700807) and by the Kazan Federal University Strategic Academic Leadership Program.

References

[1] Solovyev, V. Concreteness/Abstractness Concept: State of the Art. In: Advances in Cognitive Research, Artificial Intelligence and Neuroinformatics, pp. 275-283, Springer International Publishing, Cham, 2021

[2] Borghi, A., Binkofski, F., Castelfranchi, C., Cimatti, F., Scorolli, C., Tummolini, L. The challenge of abstract concepts, Psychological bulletin, V. 143, N. 3, 2017, pp. 263-292

[3] Brysbaert, M., Warriner, A. B., Kuperman, V. Concreteness ratings for 40 thousand generally known English word lemmas, Behavior research methods, V. 46, N. 3, 2014, pp. 904-911, doi: 10.3758/s13428-013-0403-5

[4] Brysbaert, M., Stevens, M., De Deyne, S., Voorspoels, W., Storms, G. Norms of age of acquisition and concreteness for 30,000 Dutch words, Acta Psychologica, V. 150, 2014, pp. 80-84, doi: 10.1016/j.actpsy.2014.04.010

[5] Solovyev, V. D., Ivanov, V. V., Akhtiamov, R. B. Dictionary of Abstract and Concrete Words of the Russian Language: A Methodology for Creation and Application, Journal of Research in Applied Linguistics, V. 10, 2019, pp. 218-230, doi: 10.22055/rals.2019.14684

[6] Spreen, O., Schulz, R. W. Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns, Journal of Verbal Learning and Verbal Behavior, V. 5, N. 5, 1966, pp. 459-468, doi: 10.1016/S0022-5371(66)80061-0

[7] Schmid, H.-J. English Abstract Nouns as Conceptual Shells, De Gruyter Mouton, Berlin, Boston, 2012, doi: 10.1515/9783110808704

[8] Hollis, G., Westbury, C., Lefsrud, L. Extrapolating human judgments from skip-gram vector representations of word meaning, Quarterly Journal of Experimental Psychology, V. 70, N. 8, 2017, pp. 1603-1619, doi: 10.1080/17470218.2016.1195417

[9] Coltheart, M. The MRC psycholinguistic database, The Quarterly Journal of Experimental Psychology Section A, V. 33, N. 4, 1981, pp. 497-505, doi: 10.1080/14640748108400805

[10] Rabinovich, E., Sznajder, B., Spector, A., Shnayderman, I., Aharonov, R., Konopnicki, D. and Slonim, N. Learning Concept Abstractness UsingWeak Supervision. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4854-4859, Association for Computational Linguistics, Brussels, Belgium, 2018, doi:10.18653/v1/D18-1522

[11] Feng, S., Cai, Z., Crossley, S., McNamara, D. Simulating human ratings on word concreteness. In: Proceedings of the 24th International Florida Artificial Intelligence Research Society, FLAIRS – 24, pp. 245-250, AAAI, Florida, 2011

[12] Turney, P., Neuman, Y., Assaf, D., Cohen, Y. Literal and Metaphorical Sense Identification through Concrete and Abstract Context. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 680-690, Association for Computational Linguistics, Edinburgh, Scotland, UK, 2011, https://aclanthology.org/D11-1063

[13] Tsvetkov, Y., Mukomel, E., Gershman, A. Cross-Lingual Metaphor Detection Using Common Semantic Features. In: Proceedings of the First Workshop on Metaphor in NLP, pp. 45-51, Association for Computational Linguistics, Atlanta, Georgia, 2013, https://aclanthology.org/W13-0906

[14] Huang, E., Socher, R., Manning, C., Ng, A. Improving Word Representations via Global Context and Multiple Word Prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 873-882, Association for Computational Linguistics, Jeju Island, Korea, 2012, https://aclanthology.org/P12-1092

[15] Mandera, P., Keuleers, E., Brysbaert, M. How useful are corpus-based methods for extrapolating psycholinguistic variables? Quarterly Journal of Experimental Psychology, V. 68, N. 8, 2015, pp. 1623-1642, doi: 10.1080/17470218.2014.988735

[16] Snefjella, B., Gnreux, M., Kuperman, V. Historical evolution of concrete and abstract language revisited, Behavior research methods, V. 51, N. 4, 2019, pp. 1693-1705, doi:10.3758/s13428-018-1071-2

[17] Hamilton, W. L., Clark, K., Leskovec, J., Jurafsky, D. Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 595-605, Association for Computational Linguistics, Austin, Texas, 2016, doi: 10.18653/v1/D16-1057

[18] Ljubeic, N., Fier, D., Peti-Stantic, A. Predicting Concreteness and Imageability of Words Within and Across Languages via Word Embeddings. In: Proceedings of The Third Workshop on Representation Learning for NLP, pp. 217-222, Association for Computational Linguistics, Melbourne, Australia, 2018, doi:10.18653/v1/W18-3028

[19] Charbonnier, J., Wartena, C. Predicting Word Concreteness and Imagery. In: Proceedings of the 13th International Conference on Computational Semantics - Long Papers, pp. 176-187, Association for Computational Linguistics, Gothenburg, Sweden, 2019, doi:10.18653/v1/W19-0415

[20] Friendly, M., Franklin, P., Hoffman, D., Rubin, D. The Toronto Word Pool: Norms for imagery, concreteness, orthographic variables, and grammatical usage for 1,080 words, Behavior Research Methods & Instrumentation, V. 14, 1982, pp. 375-399

[21] Paivio, A., Yuille, J., Madigan, S. Concreteness, imagery, and meaningfulness values for 925 nouns, Journal of experimental psychology, V. 76, N. 1, 1968, Suppl:pp. 1-25, doi:10.1037/h0025327

[22] Theijssen, D., van Halteren, H., Boves, L., Oostdijk, N. On the difficulty of making concreteness concrete, Computational Linguistics in the Netherlands Journal, V. 1, 2011, pp. 61-77

[23] Thompson, B., Lupyan, G. Automatic estimation of lexical concreteness in 77 languages. In: Proceedings of the 40th Annual Conference of the Cognitive Science Society (CogSci 2018), pp. 1122-1127, Cognitive Science Society, Madison, WI, USA, 2018

[24] Wang, X., Su, C., Chen, Y. A Method of Abstractness Ratings for Chinese Concepts. In: UKCI: UKWorkshop on Computational Intelligence, pp. 217-226, Springer International Publishing, 2018, doi: 10.1007/978-3-319-97982-3

[25] Dadras, P., Ramezani, M. CODAC: Concreteness Degree Auto-Calculator of Persian Words, International Journal of Computer Science and Information Security (IJCSIS), V. 15, N. 5, 2017, pp. 64-72

[26] Solovyev, V., Ivanov, V. Automated Compilation of a Corpus-Based Dictionary and Computing Concreteness Ratings of Russian. In: Speech and Computer, pp. 554-561, Springer International Publishing, Cham, 2020

[27] Harris, Z. S. Papers in structural and transformational linguistics, Reidel, Dordrecht, 1970

[28] Weeds, J., Weir, D., McCarthy, D. Characterising Measures of Lexical Distributional Similarity. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 1015-1021, COLING, Geneva, Switzerland, 2004, https://aclanthology.org/C04-1146

[29] Pantel, P. Inducing Ontological Co-Occurrence Vectors. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 05, pp. 125-132, Association for Computational Linguistics, USA, 2005, doi: 10.3115/1219840.1219856

[30] Mikolov, T., Chen, K., Corrado, G. and Dean, J. Efficient Estimation of Word Representations in Vector Space, arXiv preprint, arxiv:1301.3781, 2013, http://arxiv.org/abs/1301.3781

[31] Tang, X. A state-of-the-art of semantic change computation, Natural Language Engineering, V. 24, N. 5, 2018, pp. 649-676, doi:10.1017/S1351324918000220

[32] Tahmasebi, N., Borin, L., Jatowt, A. Survey of computational approaches to lexical semantic change detection. In: Computational approaches to semantic change, pp. 1-91, Language Science Press, Berlin, 2021

[33] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jgou, H. and Mikolov, T. FastText.zip: Compressing text classification models, arXiv preprint, arXiv:1612.03651, 2016, https://arxiv.org/abs/1612.03651

[34] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A. Advances in Pre-Training Distributed Word Representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018

[35] Xu, Y. and Kemp, C. A Computational Evaluation of Two Laws of Semantic Change, Cognitive Science, 2015

[36] Khristoforov, S., Bochkarev, V., Shevlyakova, A. Recognition of Parts of Speech Using the Vector of Bigram Frequencies. In: Analysis of Images,

Social Networks and Texts, pp. 132-142, Communications in Computer and Information Science, Springer International Publishing, Cham, V. 1086, 2020, doi: 10.1007/978-3-030-39575-9 13

[37] Solovyev, V. D., Bochkarev, V. V., Khristoforov, S. V. Generation of a dictionary of abstract/concrete words by a multilayer neural network, Journal of Physics: Conference Series, V. 1680, 2020, 012046, doi:10.1088/1742-6596/1680/1/012046

[38] Lin, Y., Michel, J.-B., Aiden Lieberman, E., Orwant, J., Brockman, W., Petrov, S. Syntactic Annotations for the Google Books NGram Corpus. In: Proceedings of the ACL 2012 System Demonstrations, pp. 169-174, Association for Computational Linguistics, Jeju Island, Korea, 2012, https://aclanthology.org/P12-3029

[39] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernndez, L. Syntactic Dependency-Based N-grams as Classification Features. In: Advances in Computational Intelligence, pp. 1-11, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013

[40] Hughes, J. M., Foti, N. J., Krakauer, D. C., Rockmore, D. N. Quantitative patterns of stylistic influence in the evolution of literature, Proceedings of the National Academy of Sciences, 2012, doi:10.1073/pnas.1115407109

[41] Savinkov, A., Bochkarev, V., Shevlyakova, A., Khristoforov, S. Neural Network Recognition of Russian Noun and Adjective Cases in the Google Books Ngram Corpus. In: Speech and Computer, SPECOM 2021, pp. 626-637, Lecture Notes in Computer Science, Springer International Publishing, Cham, V. 12997, 2021, doi: 10.1007/978-3-030-87802-3 56

[42] Bullinaria, J. A., Levy, J. P. Extracting semantic representations from word co-occurrence statistics: A computational study, Behavior Research Methods, V. 39, N. 3, 2007, pp. 510-526, doi:10.3758/BF03193020

[43] Haykin, S. Neural Networks: A Comprehensive Foundation, (2nd ed.), Prentice Hall PTR, USA, 1998

[44] Shah, A., Kadam, E., Shah, H., Shinde, S. Deep Residual Networks with Exponential Linear Unit, arXiv preprint, arXiv:1604.04112, 2016, http://arxiv.org/abs/1604.04112

[45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, J. Mach. Learn. Res., V. 15, N. 1, 2014, pp. 1929-1958

[46] Kingma, D. P., Ba, J. Adam: A Method for Stochastic Optimization, arXiv preprint, arXiv:1412.6980, 2014, http://arxiv.org/abs/1412.6980

[47] You, K., Long, M., Jordan, M. I. How Does Learning Rate Decay Help Modern Neural Networks, arXiv preprint, arXiv:1908.01878, 2019, https://arxiv.org/abs/1908.01878

[48] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai J., Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32, pp. 8024-8035, Curran Associates, Inc., 2019

[49] Kendall, M. G., Stuart, A. The advanced theory of statistics. Inference and relationship. (3rd ed.), Griffin, London, 1961

[50] Jamrozik, A., Gentner, D. Making sense of the abstract uses of the prepositions in and on. In: Proceedings of the 36th Annual Conference of the Cognitive Science Society, pp. 2411-2416, Cognitive Science Society, 2014

[51] Steen, G. J., Dorst, A. G., Herrmann, J. B., Kaal, A. A., Krennmayr, T., Pasma, T. A method for linguistic metaphor identification. From MIP to MIPVU., Converging Evidence in Language and Communication Research, John Benjamins, V. 14, 2010

[52] Hill, F., Korhonen, A. Concreteness and Subjectivity as Dimensions of Lexical Meaning. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 725-731, Association for Computational Linguistics, Baltimore, Maryland, 2014, doi: 10.3115/v1/P14-2118