Analysing the semantic content of static Hungarian embedding spaces

Tamás Ficsor[1], Gábor Berend[1,2]

[1] Institute of Informatics, University of Szeged, Hungary
[2] MTA-SZTE Research Group on Artificial Intelligence
{ficsort,berendg}@inf.u-szeged.hu

Abstract. Word embeddings can encode semantic features and have achieved many recent successes in solving NLP tasks. Although word embeddings have high success on several downstream tasks, there is no trivial approach to extract lexical information from them. We propose a transformation that amplifies desired semantic features in the basis of the embedding space. We generate these semantic features by a distant supervised approach, to make them applicable for Hungarian embedding spaces. We propose the Hellinger distance in order to perform a transformation to an interpretable embedding space. Furthermore, we extend our research to sparse word representations as well, since sparse representations are considered to be highly interpretable.

Keywords: Interpretability, Semantic Transformation, Word Embeddings

1 Introduction

Continuous vectorial word representations are routinely employed as the inputs of various NLP models such as named entity recognition (Seok et al., 2016), part of speech tagging (Abka, 2016), question answering (Shen et al., 2015), text summarization (Mohd et al., 2020), dialog systems (Forgues et al., 2014) and machine translation (Zou et al., 2013).

Static word representations acquire their lexical knowledge from local or global contexts. GloVe (Pennington et al., 2014a) uses global co-occurrence statistics to determine a word's representation in the continuous space, whereas Mikolov et al. (2013) proposed a predictive model for predicting target words from their contexts. Furthermore, Bojanowski et al. (2017) presented a training technique of word representations where sub-word information is in the form of character $n-$grams are also considered. The outputs of these word embedding algorithms are able to encode semantic relations between words (Pennington et al., 2014a; Nugaliyadde et al., 2019). This can be present on word-level – such as similarity in meaning, word analogy, antonymic relation – or word embeddings can be utilized to produce sentence-level embeddings, which shows that word vectors still carry intra-sentence information (Kenter and de Rijke, 2015).

Despite the successes of word embeddings on semantics related tasks, we have no direct knowledge of the human-interpretable information contents of dense dimensions. Utilizing human-interpretable features as prior information could lead to performance gain in various NLP tasks. Identifying and understanding the dense representation in each dimension can be cumbersome for humans. To alleviate this problem, we propose a transformation where we map existing word representations into a more interpretable space, where each dimension is supposed to be responsible for encoding semantic information from a predefined set of semantic inventory. There are various ways to form groups of semantic classes by forming semantically coherent groups of words. In this work, we shall rely on ConceptNet (Speer et al., 2016) to do so.

We measure the information contents of each dimension in the original embedding space towards a predefined set of human interpretable concepts. Our approach is inspired by Şenel et al. (2018) which utilized the Bhattacharyya distance for the aforementioned purpose. In this work, we also evaluate a close variant of the Bhattacharyya distance, the Hellinger distance for transforming word representations in a way that the individual dimensions have a more transparent interpretation.

Feature norming studies have revealed that humans usually tend to describe the properties of objects and concepts with a limited number of sparse features (McRae et al., 2005). This kind of sparse representation became a major part of natural language processing since we can see the resemblance between sparse features and human feature descriptions. Hence, we additionally explore the effects of applying sparse word representations as an input to our algorithm which makes the semantic information stored along the individual dimensions more explicit. We published our work on GitHub for interpretable word vector generation: `https://github.com/ficstamas/word_embedding_interpretability`, and shared the code for semantic category generation as well, alongside with the used semantic categories: `https://github.com/ficstamas/multilingual_semantic_categories`.

2 Related Work

Turian et al. (2010) was one of the first providing a comparison of several word embedding methods and showed that incorporating them into established NLP pipelines can also boost their performance. word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014b) and Fasttext (Bojanowski et al., 2017) methods are well known models for obtaining context-insensitive (or static) word representations. These methods generate static word vectors, i.e. every word form gets assigned a single vector that applies to all of its occurrences and senses.

The intuition behind sparse vectors is related to the way humans interpret features, which was shown in various feature norming studies (Garrard et al., 2001; McRae et al., 2005). Additionally, generating sparse features (Kazama and Tsujii, 2003; Friedman et al., 2008; Mairal et al., 2009) has proved to be useful in several areas of NLP, including POS tagging (Ganchev et al., 2010), text classification (Yogatama and Smith, 2014) and dependency parsing (Martins et al., 2011). Berend (2017) also showed that sparse representations can outperform their

	Ours	SemCat	HyperLex
Number of Categories	91	110	1399
Number of Unique Words	2760	6559	1752
Average Word Count per Category	68	91	2
Standard Deviation of Word Counts	52	56	3

Table 1. Basic statistics about the semantic categories.

dense counterparts in certain NLP tasks, such as NER, or POS tagging. Murphy et al. (2012) proposed Non-Negative Sparse Embedding to learn interpretable sparse word vectors, Park et al. (2017) showed a rotation based method and Subramanian et al. (2017) suggested an approach using a denoising k-sparse auto-encoder to generate interpretable sparse word representations. Balogh et al. (2019) made prior research about the semantic overlap of the generated vectors with a human commonsense knowledgebase and found that substantial semantic content is captured by the bases of sparse embedding space.

Şenel et al. (2018) showed a method where they measured the interpretability of the dense GloVe embedding space, and later showed a method to manipulate and improve the interpretability of a given static word representation (Şenel et al., 2020).

Our proposed approach also relates to the application of the Hellinger distance, which has been used in NLP for constructing word embeddings Lebret and Collobert (2014). Note that the way we apply the Hellinger distance differs from prior work in that we use it for amplifying the interpretability of contextual word representations, whereas the Hellinger distance served as the basis for constructing (static) embeddings in earlier work.

3 Data

3.1 Semantic Categories

Amplifying and understanding the semantic contents from word embedding spaces is the main objective of this study. To provide meaningful interpretation to each dimension, we rely on the base concept of distributional semantics (Harris, 1954; Boleda, 2020). In order to investigate the underlying semantic properties of word embeddings, we have to define some kind of semantic categories that represent the semantic properties of words. These semantic properties can represent any arbitrary relation which makes sense from a human perspective, for example, words such as *"red"*, *"green"*, and *"yellow"* can be grouped under the **"color"** semantic category which represents a hypernym-hyponym relation, but they can be found among *"traffic"* related terms as well. Another example is **"car"** semantic category which is in meronymy relation with words such as *"engine"*, *"wheels"* and *"crankcase"*.

Previous similar linguistic resources that contain semantic categorization of words include HyperLex (Véronis, 2004) and SemCat (Şenel et al., 2018). A

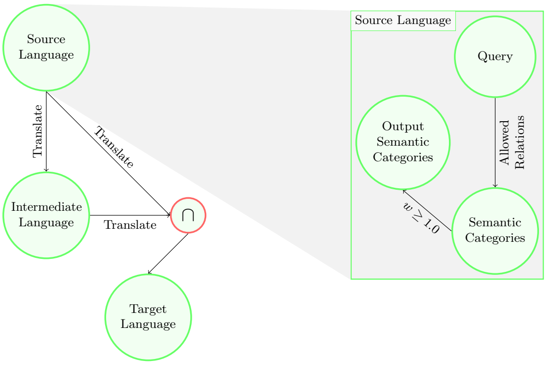

Fig. 1. Generation of semantic categories with the help of allowed relations from ConceptNet, where the Query represents the root concept, and w denotes the weight of the relation.

major problem with them from the standpoint of applicability is that these datasets are restricted to English, so they can not be utilized in multilingual scenarios. From an informational standpoint, HyperLex with a low average and standard deviation category sizes also raises concerns. In order to extend it to the Hungarian language as well, we used the semantic category names from SemCat and defined relations on a category-by-category base manually. We relied on a subset of relations from ConceptNet (Speer et al., 2016). To obtain higher quality semantic categories, we introduced an intermediate language that works as a validation to reduce undesired translations. The whole process can be followed in Figure 1.

First, we generate the semantic categories from the source language by the allowed relations and restricted the inclusion of words by the weight of the relation. Semantic category names from SemCat were used as the input (Query) and the weight of each relation is originated from ConceptNet. Then we translate the semantic categories to the target language directly and through the intermediate language to the target language, where we kept the intersection of the two results. It is recommended to rely on one of the **core** languages defined in ConceptNet as Source and Intermediate language. Using ConceptNet for inducing the semantic categories for our experiments makes it easy to extend our experiments later for additional languages beyond Hungarian. We present some basic statistics about the mentioned semantic categories in Table 1. This kind of distant supervised generation (Mintz et al., 2009) can produce large number of data easily but it carries the possibility that the generated data is noisy.

3.2 Word Embeddings

We conducted our experiments on 3 embedding spaces trained using the Fasttext algorithm (Bojanowski et al., 2017). The 3 embedding spaces that we relied on were the Hungarian Fasttext (Fasttext HU) embeddings pre-trained on

Wikipedia[3], its aligned variant[4] (Fasttext Aligned) that was created using the RCSLS criteria (Joulin et al., 2018) with the objective to bring Hungarian embeddings closer to semantically similar English embeddings and the Szeged Word Vectors (Szeged WV) (Szántó et al., 2017) which is based on the concatenation of multiple Hungarian corpora.

We limited the word embeddings to their 50,000 most frequent tokens and evaluated every experiment with this subset of all vectors. The vocabulary of the Fasttext HU and Fasttext Aligned embeddings are identical, however, it is important to emphasize that the Szeged WV overlap with the vocabulary of these embedding spaces on less than half of the word forms, i.e. 22,112 words. Furthermore, Szeged WV uses a cased vocabulary, unlike the Fasttext embeddings. In the case of Fasttext, the vocabulary of the embedding and our semantic categories overlaps in 1848 unique words. For the Szeged WV, it only overlaps with 1595 unique words.

Our approach can evaluate other embedding types as well. So due to the fact that sparse embeddings are deemed to be more interpretable compared to their dense counterparts, we also produced sparse static word representations by applying dictionary learning for sparse coding (Mairal et al., 2009) (DLSC) on the dense representation. For obtaining the sparse word representations of dense static embedding space $\mathcal{E}$, we solved the optimization problem

$$\min_{\alpha,D} \frac{1}{2} \left\|\mathcal{E} - \alpha D\right\|_F^2 + \lambda \left\|\alpha\right\|_1,$$

that is, our goal is to decompose $\mathcal{E} \in \mathbb{R}^{v\times d}$ into the product of a dictionary matrix $D \in \mathbb{R}^{k\times d}$ and a matrix of sparse coefficients $\alpha \in \mathbb{R}^{v\times k}$ with a sparsity-inducing ℓ_1 penalty on the elements of α. Furthermore, v denotes the size of the vocabulary, d represents the dimensionality of the original embedding space, and k is the number of basis vectors.

We obtained different sparse embedding space by modifying the hyperparameters of the algorithm. So we evaluated it with $\lambda \in \{0.05, 0.1, 0.2\}$ regularization and $k \in \{1000, 1500, 2000\}$ basis vectors.

4 Our Approach

4.1 Semantic Decomposition

The foundation of our approach is to measure the encoding of semantic information in the basis of pre-trained static word embeddings. In order to quantify the semantic information, we have to observe the joint behavior of similarities in semantic distributions. This approach is feasible due to distributional semantics (Boleda, 2020), which states that similarity in meaning results in similarity in linguistic distribution (Harris, 1954). This behavior can be observed from the

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.hu.vec

[4] https://dl.fbaipublicfiles.com/fasttext/vectors-aligned/wiki.hu.align.vec

fact that static word representations are trained on co-occurrence information of word tokens. So if we are able to measure the dissimilarity between a distribution that represents a semantic information and the distribution of space (which is the complementary distribution of semantic information) then we can give a transformation that is going to explicitly express the semantic categories in each dimension.

In other words, the coefficients of a dimension form a distribution $\mathcal{R} \in \mathbb{R}^v$. The desired semantic information we try to express is denoted as $\mathcal{P} \subseteq \mathcal{R}$. For example, $\mathcal{P}$ describes the *"wave"* semantic information, then words related to that term should occur in a similar context, such as *"rising"*, *"golden"*, or *"lacy"* in *"_ waves"*. So by expressing how far this distribution is from the distribution of a dimension, then we can see how significant is the dimension about the semantic information. The certainty of such a dimension about the desired semantic information can be formulated as $D(\mathcal{P}, \overline{\mathcal{P}})$. If this distance is low then it means that the information gain would be really low because the two distributions are nearly homogeneous. Analogously, if the distance is high then we can rely on that dimension with higher certainty. So the distance expresses the certainty we have in each dimension about the semantic information.

In order to express the certainty in a dimension, first, we have to separate the coefficients in a dimension to represent the previously defined distributions. As a reminder, we denoted the embedding space with $\mathcal{E}$, then we denote the defined semantic categories as $\mathcal{S}$. So we can define function $f : x \to \mathcal{E}$ which returns the representation of word token x, and function $S : x \to \mathcal{S}$ which maps word token x to its corresponding semantic category. Then we can separate the coefficients along the ith dimension and jth semantic category as

$$P_{ij} = \left\{ f(x)^{(i)} \mid f(x) \in \mathcal{E},\, S(x) \in \mathcal{S}^{(j)} \right\}$$

and similarly

$$Q_{ij} = \left\{ f(x)^{(i)} \mid f(x) \in \mathcal{E},\, S(x) \notin \mathcal{S}^{(j)} \right\},$$

where P_{ij} represents the distribution of a particular semantic category in a dimension (in-category words) and Q_{ij} $(= \overline{P_{ij}})$ represents the distribution of the rest of the dimension (out-of-category words).

4.2 Measuring dissimilarity

To measure the dissimilarity, hence observe the certainty of semantic categories in each dimension we define two distances. We apply Bhattacharyya distance as a baseline from Şenel et al. (2018) and Hellinger distance as an alternative improvement. Both distances can be expressed by Bhattacharrya coefficient (or fidelity coefficient) as

$$D_B(p,q) = -\ln \int\limits_{-\infty}^{\infty} \sqrt{p(x)q(x)}\; dx \qquad D_H(p,q) = \sqrt{1 - \int\limits_{-\infty}^{\infty} \sqrt{p(x)q(x)}\; dx},$$

where the integrand expresses the fidelity coefficient.The important differences between the two types of distances are that

- Hellinger distance is a bounded metric that eases the interpretation of values when the fidelity is close to 0,
- Hellinger distance accumulates small distributional differences better which means if the fidelity is close to 1, it can still enhance potentially significant information.

To maintain consistency, comparability and a baseline, we define Bhattacharyya distance as Şenel et al. (2018), and Hellinger distance by their closed forms which assumes normality of the investigated distributions. Under the normality assumption, the Bhattacharyya distance can be expressed as

$$D_B(P_{i,j}, Q_{i,j}) = \frac{1}{4} \ln \left(\frac{1}{4} \left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} + 2 \right) \right) + \frac{1}{4} \left(\frac{(\mu_p - \mu_q)^2}{\sigma_p^2 + \sigma_q^2} \right), \quad (1)$$

and Hellinger distance can be formulated as

$$D_H(P_{i,j}, Q_{i,j}) = \sqrt{1 - \sqrt{\frac{2\sigma_p \sigma_q}{\sigma_p^2 + \sigma_q^2}} e^{-\frac{1}{4} \cdot \frac{(\mu_p - \mu_q)^2}{\sigma_p^2 + \sigma_q^2}}}, \quad (2)$$

where σ denotes the standard deviation and μ denotes the mean of $P_{i,j}$ and $Q_{i,j}$ respectively, assuming that $P_{i,j} \sim \mathcal{N}(\sigma_p, \mu_p)$ and $Q_{i,j} \sim \mathcal{N}(\sigma_q, \mu_q)$. We then define $\mathcal{W}_D \in \mathbb{R}^{d \times |\mathcal{S}|}$ that contains the distances of semantic category-dimension pairs, i.e. $\mathcal{W}_D(i,j) = D(P_{ij}, Q_{ij})$, with D denoting either of the Bhattacharyya or Hellinger distances.

4.3 Interpretable Word Vector Generation

In order to obtain interpretable word vectors, we have to first refine the quality of transformation. It is highly possible that our semantic category dataset is imbalanced and/or during the pre-training process we do not have enough information about a word token. So we should reduce the bias of dominant semantic categories which can be obtained by performing ℓ_1 normalization on $\mathcal{W}_D$ in such a manner that the values corresponding to each semantic category sum up to 1. We shall denote the transformation matrix that we derive in such a manner as $\mathcal{W}_{ND}$.

Another problem which occurs in embedding spaces is that semantic information can be encoded in both positive and negative direction relative to the mean, hence we should adjust the orientation of these vectors in certain bases in order to couple semantic categories in their corresponding bases and segregate them from others in other bases. We determine the directions from the sign of difference between the mean of the original distributions, thus we can obtain $\mathcal{W}_{NSD}$ as

$$\mathcal{W}_{\mathcal{NSD}}(i,j) = sign(\varDelta_{ij}) \cdot \mathcal{W}_{\mathcal{ND}}(i,j),$$

where $\varDelta_{ij} = \mu_{p_{ij}} - \mu_{q_{ij}}$ and $sign$ is the signum function.

We also standardize $\mathcal{E}$ in order to avoid multicollinear issues, thus we can yield higher quality word vectors. We denote the standardized embedding space by $\mathcal{E}_S$. As a final step, we obtain our interpretable representations $\mathcal{I} \in \mathbb{R}^{v \times |\mathcal{S}|}$ as the product of $\mathcal{E}_S$ and $\mathcal{W}_{NSD}$.

5 Evaluation methods

5.1 Word Retrieval Test

We are concerned about the accuracy of our model, to know how well it behaves on unknown data. In $\mathcal{W}_\mathcal{D}$ we can see the semantic distribution of the dimensions and in $\mathcal{I}$ each column should represent a semantic category. So each dimension in $\mathcal{I}$ should ideally represent a semantic category from the semantic categories.

In order to measure the semantic quality of $\mathcal{I}$, we used 60% of the words from each semantic category for training and 40% for evaluation. By relying on the training set, we calculate the distance matrix $\mathcal{W}_\mathcal{D}$ from the embedding space, using any arbitrary distance we defined earlier. We also experiment with a pruned version of $\mathcal{W}_\mathcal{D}$ by keeping the highest $\mathcal{K}$ coefficients for each semantic category and setting the rest to 0, and denoting it as $\mathcal{W}_\mathcal{D}^\mathcal{S}$. We do that, so we can inspect the importance of the strongest encoding dimensions. Then by employing $\mathcal{W}_\mathcal{D}^\mathcal{S}$ instead of $\mathcal{W}_\mathcal{D}$, we do everything in the same way as we defined earlier.

We use the validation set and see whether the words of a semantic category are seen among the top n, $3n$ or $5n$ words in the corresponding dimension in $\mathcal{I}_\mathcal{S}$, where n is the number of the words in the validation set varying across the semantic categories. The final accuracy is calculated as the weighted mean of the accuracy of the dimensions, where the weight is the number of words in each category for the corresponding dimension.

5.2 Interpretability

In order to measure the interpretability of the semantic space, we use a functionally-grounded evaluation method (Doshi-Velez and Kim, 2017), which means it does not involve humans in the process of quantification. Furthermore, we use continuous values to express the level of interpretability (Murdoch et al., 2019).

The metric we rely on is an adaptation of the one proposed in (Şenel et al., 2018). We ought to have a metric that is independent from the dimensionality of the embedding space, so models with different number of dimensions can be compared more meaningfully.

$$IS^+_{i,j} = \frac{|S_j \cap V_i^+(\beta \times n_j)|}{n_j} \quad (3) \qquad IS^-_{i,j} = \frac{|S_j \cap V_i^-(\beta \times n_j)|}{n_j} \quad (4)$$

Eqn. (3) and (4) define the interpretability score for the positive and negative directions, respectively. In both equation i represents the dimension ($i \in \{1, 2, 3, \ldots, d\}$, where d is the number of dimensions of the embedding space) and j the semantic categories ($j \in \{1, 2, 3, \ldots, c\}$, where c is the number of the semantic categories). S_j represents the set of words belonging to the jth semantic category, n_j the number of words in that semantic category. V_i^+ and V_i^- gives us the top and bottom words selected by the magnitude of their coordinates

	Hellinger			Bhattacharyya		
β	1	5	10	1	5	10
Fasttext HU	22.00	38.43	46.87	21.29	38.80	47.01
Fasttext Aligned	**26.81**	**43.71**	**51.26**	**25.92**	**43.45**	**51.22**
Szeged WV	16.34	31.71	40.04	15.69	31.50	39.91

Table 2. Interpretability of Hungarian Fasttext, Aligned Fasttext and Szeged WV with different β relaxation and applied distance.

respectively along the ith dimension. $\beta \times n_j$ is the number of words selected from the top and bottom words, hence $\beta \in \mathbb{N}^+$ is the relaxation coefficient, as it controls how strict we measure the interpretability. As the interpretability of a dimension-category pair, we take the maximum of the positive and negative direction according to

$$IS_{i,j} = \max\left\{IS^+_{i,j}, IS^-_{i,j}\right\}. \tag{5}$$

Once we have the overall interpretability ($IS_{i,j}$), we calculate the categorical interpretability according to Eqn. (6). Şenel et al. (2018) took a different approach of taking the average of the maximum values over all the categories, however, this could easily overestimate the true interpretability of the embedding space.

In order to avoid the overestimation of the interpretability of the embedding space, we calculate Eqn. (6), where we have a condition on the selected i dimension which is defined by Eqn. (7). It chooses the highest encoding dimension according to $\mathcal{W}_D$ (distance matrix of the examined space) which ensures that we obtain the interpretability score from the most likely encoding dimension. This method is more suitable to obtain the interpretability scores, because it relies on the distribution of the semantic categories, instead of the interpretability score equally sampled from each dimension.

$$IS_j = IS_{i^*_j,j} \times 100 \qquad (6) \qquad\qquad i^*_j = \arg\max_{i'} \mathcal{W}_\mathcal{D}(i', j) \qquad (7)$$

Finally, we define the overall interpretability of the embedding space by taking the average of the interpretability scores across the semantic categories, $IS = \frac{1}{c}\sum_{j=1}^{c} IS_j$, where c is the number of categories.

6 Results

6.1 Dense Representations

We transformed all 3 embedding spaces to their interpretable representations and measured the effectiveness of the encoding by the interpretability score which can be seen in Table 2. Furthermore, we measured the generalisability of the transformation with word retrieval test which is presented in Figure 2. These types of evaluations are better observed jointly because they represent a different aspect of the embedding space but we can not make any conclusion without each other.

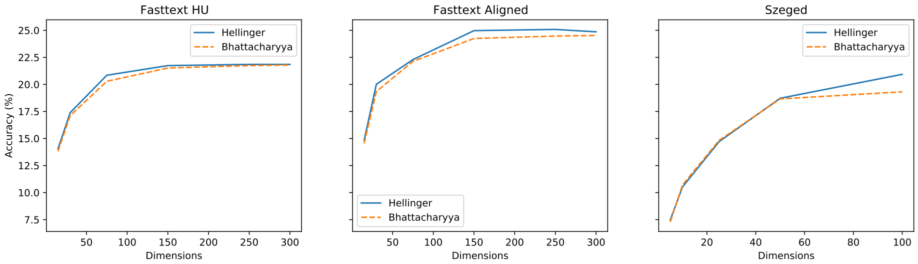

Fig. 2. The results of word retrieval tests with a relaxed size of retrieved words, where the dimensions represent the $\mathcal{K}$ kept coefficient from $\mathcal{W}_D$.

	Fasttext HU			Fasttext Aligned			Szeged WV		
λ	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
Hellinger distance									
$k = 1000$	58.11	43.21	19.33	60.13	47.58	24.25	58.88	53.85	33.82
$k = 1500$	64.49	49.24	23.82	65.50	52.20	28.44	65.03	60.94	38.14
$k = 2000$	**68.29**	52.53	26.98	**68.79**	57.05	30.63	**67.65**	64.08	42.22
Bhattacharyya distance									
$k = 1000$	53.20	33.98	18.72	55.54	37.88	22.08	56.13	45.52	27.79
$k = 1500$	57.77	36.33	21.59	59.91	39.54	24.61	62.85	50.53	30.77
$k = 2000$	**60.82**	39.03	24.43	**62.99**	42.26	26.43	**64.45**	52.18	33.12

Table 3. The effects of relying on sparse static word representation with different hyperparameters for regularization coefficient (λ) and number of basis vectors (k). Interpretability scores represented at $\beta = 1$ relaxation.

We can immediately spot the dominant performance on both evaluation methods by the aligned Fasttext word vectors. It can indicate that either the alignment could carry extra semantic knowledge or the English Wikipedia corpus is a higher quality. Szeged WV seems to be the worst-performing model according to interpretability, but it is not necessarily the case because it has a third of the number of dimensions than the Fasttext models, and differ in overlap of words in the vocabulary. In Figure 2 we can also see that it has just enough dimensions (maybe it could utilize a little bit more). This can be seen by observing the accuracy of the embedding spaces. The accuracy has not peaked before relying on all 100 of the dimensions, unlike Fasttext HU which peaks between 150 and 250 dimensions. Furthermore, it does not have a plateau-like effect where we yield little to no improvement. But these observations only apply from the standpoint of our semantic categories, not in a general manner.

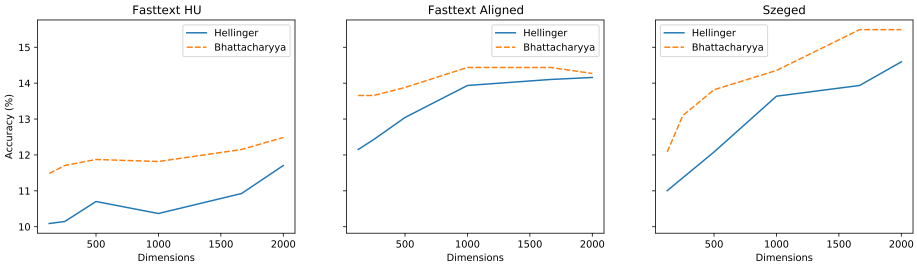

Fig. 3. The results of word retrieval tests on sparse representations ($\lambda = 0.05$ and $k = 2000$), where the dimensions represent the $\mathcal{K}$ kept coefficient from $\mathcal{W}_D$.

6.2 Sparse Representations

If we closely inspect Eqn. (1) and (2), we can see that division errors occur when σ_p or σ_q equals 0. When the standard deviation for P or Q would be 0, we replace it by $\sqrt{10^{-5}}$ instead.

We evaluated our experiments with different hyperparameters for sparse vector generation as we can see in Table 3 when using the $\beta = 1$ relaxation. We can conclude that increasing the level of sparsity does not benefit the interpretability. On the other hand, changing the number of basis vectors has a beneficial impact. We can see that sparse representation amplifies the semantic information on each basis, since the interpretability of these embedding spaces improved by 2-3 times.

Figure 3 demonstrates the results of the word retrieval test when using sparse representations obtained when setting $\lambda = 0.05$ and $k = 2000$. We can see that the generalisability of the model is decreased overall, and we should rely on more $\mathcal{K}$ none zero coefficients to extract the semantic information. This could be the cause of high level of noise is present in our semantic categories.

6.3 Semantic Decomposition

We can see the semantic decomposition of the word "ember" on Figure 4. In the first row, we represent the dense and in the second we represent the sparse embedding spaces. We expect that in this case for the "ember" word, semantic categories that contain this word are among the highest coefficients. We can see that, after we obtained the sparse representations for Fasttext, and transformed them the semantic decomposition shows an identical representation even though their scores are different.

7 Conclusion

We evaluated the transformation of non-contextual embedding spaces into a more interpretable one, which can be used to analyze the semantic distribution which can have a potential application in knowledge base completion. We investigated

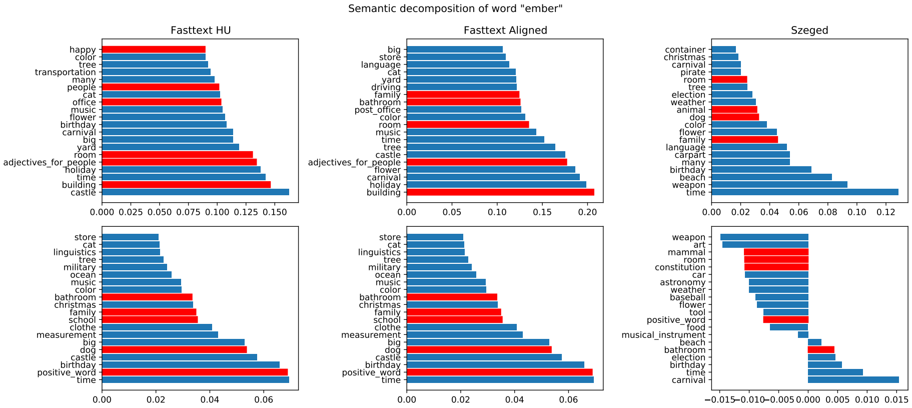

Fig. 4. Semantic decomposition of the word "ember". First row shows the decomposition of dense embedding spaces and the second represents the sparse embedding spaces ($k = 2000$, $\lambda = 0.05$). On the y axis we represent the semantic categories and on the x axis we show the corresponding weights of the word. Red bars represents that if the word is in the semantic category.

the interpretability of the Hungarian Fasttext, Hungarian Aligned Fasttext, and Szeged WV models as source embeddings, where we concluded that all of them are capable to express the anticipated semantic information contents and that the aligned word vectors performed above all. Furthermore, we proposed a modified version of the interpretability score, which let us compare the interpretability of embedding spaces with different dimensionality and consider errors from the transformation.

We also considered the utilization of the Hellinger distance instead of Bhattacharyya distance which improved the interpretability scores. Furthermore, we explored the behavior of sparse representations. As for the hyperparameter selection, we can conclude that we want to increase the number of the basis, and decrease the sparsity level in order to improve the performance.

However, if we consider sparse representations the generalisability of the embedding may decrease, but it might be a joint factor of the distant supervised generation of Hungarian semantic categories and random selection of validation test sets. If our semantic categories contain too much noise then it could accumulate that noise during the transformation which is indicated by the high interpretability score, and a lower score on the word retrieval test (which can represent a distinct distribution from the original distribution of the semantic category).

Acknowledegments

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program and the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008). The work of Tamás Ficsor was funded by the project "Integrated program for training new generation of scientists in the fields of computer science", no EFOP-3.6.3-VEKOP-16-2017-0002, supported by the EU and co-funded by the European Social Fund.

Bibliography

Abka, A.: Evaluating the use of word embeddings for part-of-speech tagging in bahasa indonesia. pp. 209–214 (10 2016)

Balogh, V., Berend, G., Diochnos, D.I., Turán, Gy.: Understanding the semantic content of sparse word embeddings using a commonsense knowledge base (2019)

Berend, G.: Sparse coding of neural word embeddings for multilingual sequence labeling. Transactions of the Association for Computational Linguistics 5, 247–261 (2017), `https://www.aclweb.org/anthology/Q17-1018`

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)

Boleda, G.: Distributional semantics and linguistic theory. Annual Review of Linguistics 6(1), 213–234 (Jan 2020), `http://dx.doi.org/10.1146/annurev-linguistics-011619-030303`

Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017)

Forgues, G., Pineau, J., Larchevêque, J.M., Tremblay, R.: Bootstrapping dialog systems with word embeddings. In: Nips, modern machine learning and natural language processing workshop. vol. 2 (2014)

Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics (Oxford, England) 9, 432–41 (08 2008)

Ganchev, K., Graça, J.a., Gillenwater, J., Taskar, B.: Posterior regularization for structured latent variable models. J. Mach. Learn. Res. 11, 2001–2049 (Aug 2010)

Garrard, P., Ralph, M., Patterson, K.: Prototypicality, distinctiveness, and intercorrelation: Analyses of the semantic attributes of living and nonliving concepts. Cognitive neuropsychology 18, 125–74 (03 2001)

Harris, Z.S.: Distributional structure. WORD 10(2-3), 146–162 (1954), `https://doi.org/10.1080/00437956.1954.11659520`

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., Grave, E.: Loss in translation: Learning bilingual word mapping with a retrieval criterion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018)

Kazama, J., Tsujii, J.: Evaluation and extension of maximum entropy models with inequality constraints pp. 137–144 (01 2003)

Kenter, T., de Rijke, M.: Short text similarity with word embeddings. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1411–1420. CIKM '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2806416.2806475

Lebret, R., Collobert, R.: Word embeddings through hellinger pca. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (2014)

Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. vol. 382, p. 87 (01 2009)

Martins, A., Smith, N., Figueiredo, M., Aguiar, P.: Structured sparsity in structured prediction. pp. 1500–1511 (01 2011)

McRae, K., Cree, G., Seidenberg, M., Mcnorgan, C.: Semantic feature production norms for a large set of living and nonliving things. Behavior research methods 37, 547–59 (12 2005)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality (2013)

Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 1003–1011. Association for Computational Linguistics, Suntec, Singapore (Aug 2009), https://www.aclweb.org/anthology/P09-1113

Mohd, M., Jan, R., Shah, M.: Text document summarization using word embedding. Expert Systems with Applications 143, 112958 (2020), http://www.sciencedirect.com/science/article/pii/S0957417419306761

Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Definitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences 116(44), 22071–22080 (Oct 2019), http://dx.doi.org/10.1073/pnas.1900654116

Murphy, B., Talukdar, P., Mitchell, T.: Learning effective and interpretable semantic models using non-negative sparse embedding. In: Proceedings of COLING 2012. pp. 1933–1950. The COLING 2012 Organizing Committee, Mumbai, India (Dec 2012), https://www.aclweb.org/anthology/C12-1118

Nugaliyadde, A., Wong, K.W., Sohel, F., Xie, H.: Enhancing semantic word representations by embedding deeper word relationships. CoRR abs/1901.07176 (2019), http://dblp.uni-trier.de/db/journals/corr/corr1901.html

Park, S., Bak, J., Oh, A.: Rotated word vector representations and their interpretability. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 401–411. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017), https://www.aclweb.org/anthology/D17-1041

Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation pp. 1532–1543 (Oct 2014a)

Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014b)

Seok, M., Song, H.J., Park, C.Y., Kim, J.D., Kim, Y.S.: Named entity recognition using word embedding as a feature. International Journal of Software Engineering and its Applications 10, 93–104 (2016)

Shen, Y., Rong, W., Nan, J., Peng, B., Tang, J., Xiong, Z.: Word embedding based correlation model for question/answer matching (11 2015)

Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: AAAI Conference on Artificial Intelligence (2016), http://arxiv.org/abs/1612.03975

Subramanian, A., Pruthi, D., Jhamtani, H., Berg-Kirkpatrick, T., Hovy, E.: Spine: Sparse interpretable neural embeddings (2017)

Szántó, Z., Vincze, V., Farkas, R.: Magyar nyelvű szó-és karakterszintű szóbeágyazások. XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018), Szeged, Szegedi Tudományegyetem, Szegedi Tudományegyetem pp. 323–328 (2017)

Turian, J., Ratinov, L.A., Bengio, Y.: Word representations: A simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 384–394 (2010)

Véronis, J.: Hyperlex: lexical cartography for information retrieval. Comput. Speech Lang. 18(3), 223–252 (2004), http://dblp.uni-trier.de/db/journals/csl/csl18.html#Veronis04

Yogatama, D., Smith, N.A.: Linguistic structured sparsity in text categorization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 786–796. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014), https://www.aclweb.org/anthology/P14-1074

Zou, W.Y., Socher, R., Cer, D., Manning, C.D.: Bilingual word embeddings for phrase-based machine translation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1393–1398. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013), https://www.aclweb.org/anthology/D13-1141

Şenel, L.K., Utlu, I., Yücesoy, V., Koç, A., Çukur, T.: Semantic structure and interpretability of word embeddings. IEEE/ACM Transactions on Audio, Speech, and Language Processing 26(10), 1769–1779 (2018)

Şenel, L.K., Utlu, I., Şahinuç, F., Ozaktas, H.M., Koç, A.: Imparting interpretability to word embeddings while preserving semantic structure. Natural Language Engineering p. 1–26 (2020)