Msc Natural Language Processing 2022-2023

UE 705 - Supervised Project

Intrinsic Evaluation of Word Embeddings

Students:
Clémentine BLEUZE
Ekaterina GOLIAKOVA
Chun YANG

Supervisor:
David LANGLOIS

Contents

Introduction

Word Embeddings (WE) are vector representations of words, typically constituted by hundreds to thousands of real-valued components. Such representations are calculated on the basis of big training corpora (in the order of billions of tokens) and most often using Neural Networks (NN). The major underlying assumption made by WE models is that similar words are represented by similar vectors, in other words, it is assumed that linguistic and semantic information gets encoded in the vector space.

In recent years, WE models have known a significant development. In fact, WE progressively outperformed more traditional *word representation* approaches, such as statistical ones (often based on word-word or word-document co-occurrence matrix) in a range of NLP tasks. This increased performance explains the growing interest of the research community in better understanding WE and the way they encode information. Following the proposal of David Langlois (member of the SMarT[1] team at the LORIA), we are interested in the question of linguistic information retrieval in WE. In this project, we aim to answer the intuitively simple (yet more complex than they seem) questions working with WE models for the French language:

- *Where* does the information get encoded in WE?
- Can we locate precisely, in terms of components in the vector representation, linguistic information such as Part-of-Speech (PoS), grammatical gender, the tense of verbs, etc.?

This research will deal with the topic of the interpretability of WE. In order to answer these questions, we need to get acquainted with some of the most common WE models, which will be done in the first part of this report. Indeed, different models may not necessarily behave the same way at the components level and it is possible some of them will allow for precise retrieval of features and some others will not.

Second, investigating WE evaluation methods will provide us with useful tools for further analysis of the produced vectors. *Intrinsic evaluation* in particular tends to assess the conservation of word relationships in the vector space, which constitutes a solid ground for our feature investigation. However, we will also discuss the limits of evaluators' standards and interpretability.

Finally, we will study works related to ours. In our third part, we will focus on papers dealing with the extraction of information from WE. This can be done either by retrieving precise features, as we intend to do, or by spotting more global similarities between vectors in order to induce new linguistic knowledge.

[1]For more information, see: https://www.loria.fr/fr/la-recherche/les-equipes/smart/

1 What are Word Embeddings?

Generally speaking, *word representation* deals with the encoding of a given word w and its context c as a numerical vector $\vec{v}$, which can be later used in a variety of Natural Language Processing (NLP) tasks such as Named Entity Recognition (NER), Part-of-Speech (PoS) tagging, Word Sense Disambiguation (WSD), etc. It was proposed to classify word representation methods into the following two categories (Aliane, 2019):

- **Statistical models** or *Count-based models* rely on statistical counts of co-occurrences between words and their context.
- **Predictive models** or *Word Embeddings* (WE) use advanced learning algorithms to infer a representation for words on the basis of (word, context) information.

After saying a few words about statistical models for Word Representation, we are going to explore a few of the most common models of WE.

1.1 Statistical Models

Although they are not, technically speaking, WE methods, statistical models for *word representation* can be considered as their predecessors. These methods rely on the establishment of a co-occurrence matrix between words and their context, the latter being another word, a window of words, or sometimes even a full document.

1.1.1 One-Hot Encoding

One of the most simple count-based techniques to represent words as a vector is *One-Hot Encoding* which is still applicable to a variety of NLP tasks. The underlying idea is to create a vector of size $|V|$ (where V is the considered *Vocabulary*) filled with 0s, excepted in the position of the word in V where we'll have a 1. For example:

$$\texttt{Vocabulary = [We, love, word, embeddings]}$$

$$\texttt{We} \Longrightarrow [1, 0, 0, 0]$$

$$\texttt{love} \Longrightarrow [0, 1, 0, 0]$$

$$\texttt{word} \Longrightarrow [0, 0, 1, 0]$$

$$\texttt{embeddings} \Longrightarrow [0, 0, 0, 1]$$

This type of encoding can be used for encoding semantic tags, classification information as well as other NLP tasks, however, is pretty limiting: with a large vocabulary, a matrix of *word representations* will become too big to work with and

take too much processing resources. In addition to this, it is almost impossible to extract any meaning and compare different words using this representation.

1.1.2 Bag of Words

Another core concept of count-based word representation is *Bag of Words* (BoW) (Harris, 1954). This approach takes into account co-occurrences of words together in a corpus of text. We can illustrate this in Table 1.

```
Corpus = ["We love word embeddings", "What are word embeddings",
                        "What do you love"]
```

Table 1: Co-occurrence of words in the corpus. The rows and columns represent unique words of the corpus. The numbers on the intersections represent in how many sentences (or *documents*) the words are seen together.

	we	**love**	**word**	**embeddings**	**what**	**are**	**do**	**you**
we	0	1	1	1	0	0	0	0
love	1	0	1	1	1	0	1	1
word	1	1	0	2	1	1	0	0
embeddings	1	1	2	0	1	1	0	0
what	0	1	1	1	0	1	1	1
are	0	0	1	1	1	0	0	0
do	0	1	0	0	1	0	0	1
you	0	1	0	0	1	0	1	0

1.1.3 Popular methods of statistical word representation

Though much more complex than the techniques described above, statistical-based models still use counting principles at their core. *Latent Semantical Analysis (LSA)* is a method based on calculating word and document co-occurrences, elevating the above-mentioned BoW. Similarly based on BoW methodology, *Hyperspace Analogue to Language (HAL)* prioritizes calculating co-occurrences in a reading window of 10 words. Another statistical evaluator that relies on the co-occurrence is *Pointwise Mutual Information (PMI)* which computes a score to define collocations (2 or more words that are very likely to be used one after another).

1.2 Predictive models

The approach for word representation generation started changing from the predetermined statistical approach towards predictive models with the popularization of NN and particularly *Recurrent NNs* (RNNs) for language modeling. The NN approach was first proposed in (Bengio et al., 2003) where it was suggested to use feedforward neural networks with fixed-length context. The main disadvantage of such an approach was the aforementioned fixed-length context: NN could only use a small number of surrounding words for their context of usage.

This was addressed later (Mikolov et al., 2010a,b, 2011). In these works, it was proposed to use the RNN structure for its ability to retain the context of much longer sequences. As shown in Figure 3, the network consists of an input layer x, hidden (or context) layer s and output layer y and creates a loop of context being reused through the layers:

- Input vector $x(t)$ is calculated as $w(t) + s(t-1)$ where $w(t)$ is the vector representation of the current word and $s(t-1)$ is the context layer of the previous step.
- For each hidden layer $s(t)$ a following sigmoid activation function f is used: $s_j(t) = f(\sum_i (x_i(t)u_{ji})$
- For the output layer is calculated using a softmax function g in the following manner: $s_j(t) = g(\sum_j (s_j(t)u_{jk})$

As a result of such training, the model creates an optimal numerical *word representation* which as well encodes a large context of where and with what words the original word was used.

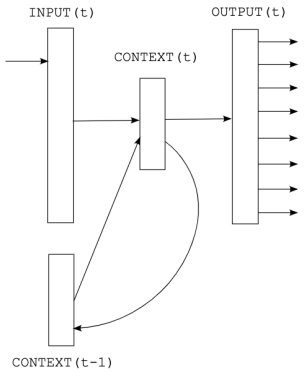

Figure 3: Representation of RNN structure (Mikolov et al., 2010a)

.

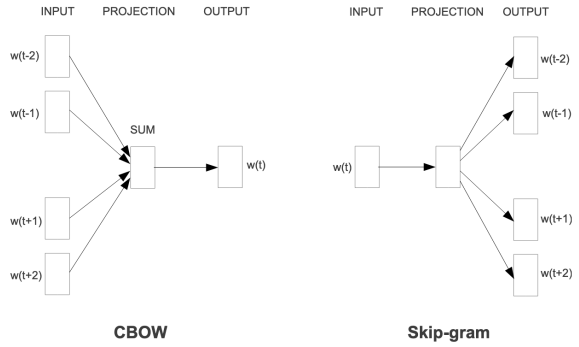

Figure 4: Architecture of CBOW and Skip-gram (Mikolov et al., 2013b)

.

1.2.1 Word2Vec: Continuous Bag of Words and Skip-gram

To elevate the idea of *word representations* generated during RNN training, the Word2Vec toolkit was proposed (Mikolov et al., 2013b,c) which actually consists of two different language models: *Continuous Bag of Words* (*CBOW*) and *Skip-gram*. Because of their low computational complexity, these models can be trained on large corpora in a short time, encoding billions of words in hours.

The main difference between the two is that CBOW's goal is to predict input words given their context, while the Skip-gram model's goal is to predict context given a target word, as can be seen in Figure 4. For *CBOW*, the output layer is shared for all words, which are projected on the same position to get the average of their vectors. The main purpose of *CBOW* is to sum the word vectors of the context and then feed the resulting vectors to a log-linear classifier to predict the target word. The model compares its predictions to expected outcomes and adjusts network parameters: vector representations of words, through gradient backpropagation. It aims at minimizing the value of the average log probability as in the formula shown below:

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-n},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+n}) \tag{1}$$

where T is the size of the vocabulary and n is the size of the context window around the target word w_t. The *Skip-gram* model, on the other hand, aims to help predict surrounding words in a sentence or document given a word w. For example, given a sequence of training words, w_1, w_2, w_3, ... , w_t , it is achieved by maximizing the

average log probability below:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+1}|w_t) \tag{2}$$

where T is the number of vocabulary words and c is the size of the training context.

Word2vec models are efficient in encoding word semantics in the vector space. *Skip-gram* performs well on the *Analogy Task* introduced by Mikolov: the vector operation $\vec{king} - \vec{man} + \vec{woman}$ produces a vector close to $\vec{queen}$ (Mikolov et al., 2013d). Intuitively, this corresponds with the fact that "*queen* is to *woman* what *king* is to *man*".

1.2.2 GloVe

The *GloVe* model (Global Vectors) was introduced later (Pennington et al., 2014). Its main contribution over previous WE models is that it captures the advantages of both (i) global matrix-factorization approaches (e.g. *LSA*) and (ii) local context-based ones (e.g. *Skip-gram*). More precisely, Pennington *et al.* show that semantic relations can be derived from a global word-word co-occurrence matrix. In particular, given two words i and j, the global matrix allows us to calculate the number of times i and j appear in the same context. This count, X_{ij}, can be used to compute the probability that j follows i, namely $P(j|i)$ as follows, where X_i is the total count of word i in the corpus:

$$P(j|i) = \frac{X_{ij}}{Xi}.$$

It is interesting to see that, using a third word k (called the *probe word*), the proximity of ratio $\frac{P(k|i)}{P(k|j)}$ to 1 indicates whether words i and j are equally close to k. This is relevant information in knowing whether i and j are similar, and here lies the intuition in the *GloVe* model: an example is provided in Figure 5. Using log-bilinear regression, *GloVe* produces word vectors such that the quantity $\vec{w_i} \cdot \vec{w_j} - ln(X_{ij})$ is minimized (where $\vec{w_i}$ and $\vec{w_j}$ are vectors for words i and j), which is demonstrated to encode the aforementioned probability-ratio information in the vector space.

GloVe vectors perform very efficiently in Mikolov's *Analogy Task*, as well as in the nearest neighbours tasks, even for rare words. Although it is stated that *GloVe* significantly outperforms Word2Vec algorithms in terms of training time and results (Pennington et al., 2014), some researchers indicate that the issue remains controversial (Aliane, 2019).

1.3 Enriching WE models

Following our research about WE models, we found valuable work interested in fine-tuning existing models. More precisely, it is possible to use pre-trained vectors

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Figure 5: Using ratios of probabilities from the co-occurrence matrix allows to deduce that *fashion* and *water* are equally close to *ice* and *steam*, whereas *gas* is closer to *steam*, and *solid* is closer to *ice* (Pennington et al., 2014)

.

embedded with *Skip-gram* or *GloVe* and guide them into encoding new information, as well as train other models for WE to encode more information.

1.3.1 Morphological Enrichment

In one of the attempts to enrich WE, an existing Log-Bilinear (LBL) language model was fine-tuned with morphological information (Cotterell and Schütze, 2015). In order to produce embeddings that encode more morphology, the authors added a task-specific objective (morphological tag prediction) to the initial word-prediction task of the language model.

With the case study of German, which is a morphologically-rich language, the authors succeed in producing vectors that are of similar quality as the pre-trained ones in reference tasks, while being more reflective of morphological information. In Figure 6, we see that words are clustered not only by PoS (*e.g* verbs are in green and adjectives in blue) but also by morphological features.

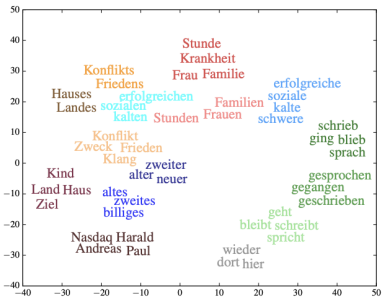

Figure 6: 2-dimensional projections of a few morphological WE. As an example, we see that past simple verbs (*schrieb, ging, blieb, sprach*) and genitive nouns (*Konflikts, Friedens*) group together (Cotterell and Schütze, 2015)

.

Considering our project to track precise information in the components of WE, we might be interested in guiding these WE towards some targeted features. We

can indeed make the hypothesis that Cotterell *et al.*' method makes morphological information more prominent into the components of the pre-trained WE.

1.3.2 Linguistic enrichment

However, contrary to this very task-specific approach, Faruqui *et al.* proposed a *post-processing* method for WE that they call *retrofitting* (Faruqui et al., 2015). Since the way the word vectors were encoded before retrofitting doesn't matter, this method can be applied to any type of WE, and not only to LBL as in (Cotterell and Schütze, 2015).

In this paper, valuable linguistic information is added to WE: synonymy, hyponymy, hyperonymy (*WordNet*[2]); paraphrase (*PPDB*[3]) and lexical + predicate-argument semantics (*FrameNet*[4]). This linguistically-rich information is used to produce a graph connecting words from the considered vocabulary. The belief propagation algorithm is then used to update the initial word vectors, which result in updated vectors that are similar to their initial version while being close to the vectors of their connected neighbors in the graph. The guided version of WE is of a higher quality than the previous one, in particular in tasks relying on the newly-encoded information, similar to (Cotterell and Schütze, 2015). In Figure 7, we see that retrofitted vectors encode analogical relationships more efficiently than non-retrofitted ones. Because of its ability to be applied to any WE, this approach might be especially interesting for the realization part of our project.

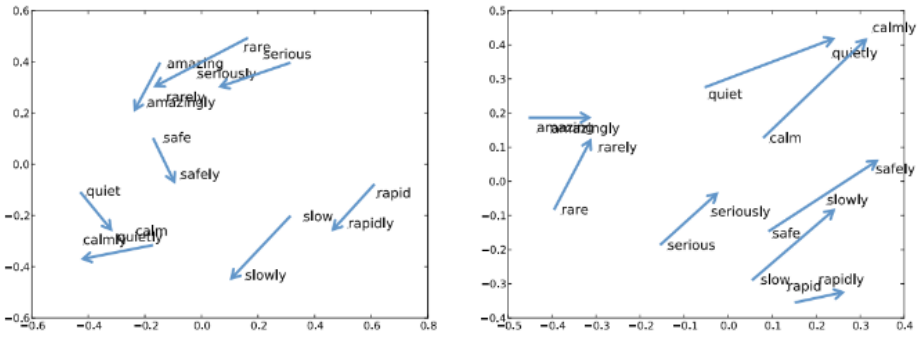

Figure 7: Two-dimensional PCA (Principal Component Analysis) projection of 100-dimensional Skip-gram vector pairs holding the "adjective to adverb" relation, before (left) and after (right) retrofitting. After retrofitting, they all point in the same direction (Faruqui et al., 2015)

.

[2]https://wordnet.princeton.edu/
[3]Paraphrase Database. See http://paraphrase.org/#/download.
[4]https://framenet.icsi.berkeley.edu/fndrupal/

2 Word Embeddings Evaluation

Since different models produce different WE, evaluation methods are needed in order to assess their differences in terms of downstream tasks performance or linguistic quality. This is done using respectively *extrinsic* and *intrinsic* evaluation.

2.1 Intrinsic Evaluation

Intrinsic evaluators assess the quality of a representation independently of any concrete NLP application (Aliane, 2019). What is investigated instead is the extent to which semantic and syntactic properties of the words can be retrieved from their embeddings.

2.1.1 Word Similarity

Word similarity evaluation consists of measuring the extent to which linguistically close words are encoded by vectors that are close in the vector space. This requires golden standards of linguistically-similar words (this similarity is established by human experts or simple evaluators). For instance, popular *word similarity* evaluator *WordSim353* contains 353 pairs of words whose similarity ranges on a scale from 0-10, whereas *MTurk* has 771 pairs and a 0-5 scale.

Next, the closeness of two vectors can be measured using cosine-similarity, which is computed as follows:

$$cos(\vec{w_x}, \vec{w_y}) = \frac{\vec{w_x}.\vec{w_y}}{\|\vec{w_x}\| * \|\vec{w_y}\|} \tag{3}$$

Intuitively, this quantity will be close to 1 for similar vectors and close to 0 for vectors pointing in very different directions. Finally, the correlation between the linguistic similarity of *gold pairs* and the cosine similarity of their vector representation is calculated: the higher the correlation, the better the score of the WE.

However, it is important to point out that similarity is an ambiguous notion. Words can be similar in different ways: they can be synonyms (*car* and *automobile*) or related (*car* and *road*), which constitutes a similarity-relatedness axis (Hill et al., 2015). Moreover, they can also be similar in terms of syntax (*sing*, *singing*) (Batchkarov et al., 2016). To address this some researchers propose these two aspects as two axes of similarity: the semantic/syntax axis and the similarity/relatedness axis (Artetxe et al., 2018). They proposed a new post-processing method to tailor a given set of embeddings. It can be generalized as a continuous parameter of the linear transformation of WE vectors, which can adjust the performance of WE in the similarity/relevance and semantic/syntactic axes without the need of additional resources.

2.1.2 Word Analogy

The *word analogy* estimator was popularized with Word2Vec models (Mikolov et al., 2013a). It assesses whether analogical relationships found in Natural Language can be found in the vector space of WE. Given a quadruplet of words *(a, b, x ,y)* such that "a is to b what x is to y", we wonder whether $\vec{a}-\vec{b} = \vec{x}-\vec{y}$ (which corresponds to the analogical relationship being conserved in the vector space). Because perfect equality might not occur despite good analogy conservation, what is actually computed is the vector $\vec{a} - \vec{b} + \vec{x}$. The closest vector to this latter is expected to be $\vec{y}$, in which case the analogy test is passed. Equations 4 and 5 display standard notation for this analogy task, as well as an example of analogical pairs.

$$a : b :: x :? \tag{4}$$

$$France : Paris :: Germany : Berlin \tag{5}$$

In (Artetxe et al., 2018), the impact of the proposed post-processing method on word analogy is evaluated. Even though post-processing can improve the accuracy of word analogy, the improvement of the accuracy of the semantic axis will reduce the accuracy of the grammatical axis, and vice versa.

2.1.3 Concept Categorization

Concept categorization can be seen as a generalized *word similarity* evaluator. On the basis of gold standard data containing categories of words (for instance *apple*, *banana* belonging to the semantical category *fruit* and *milk*, *tea* to *drink*), clusters of embeddings in the vector space are calculated using a clustering algorithm. The performance depends on the purity of these clusters: purity refers to whether each cluster contains concepts from the same or different categories (Baroni et al., 2014).

2.1.4 QVEC and QVEC-CCA

QVEC, an intrinsic estimator which was shown to be closely related to downstream tasks, was proposed as one of the concept categorization metrics (Tsvetkov et al., 2015). *QVEC* measures the correlation between word vectors from WE models and manually constructed language vectors, with the goal of maximizing the correlation with manually annotated language resources. However, *QVEC* is not a perfect estimator. First, it is not invariant to a linear variation of the WE base, however, the base in WE is usually arbitrary. Second, the scores it produces vary with the dimensionality in the embedding matrix, and thus cannot be used to compare models of different dimensions (Szegedy et al., 2014; Tsvetkov et al., 2016).

A new *internal* estimator *QVEC-CCA* was introduced to address the shortcomings of the original *QVEC* (Tsvetkov et al., 2016). The correlation between WE and manually constructed vectors is now computed employing canonical correlation

analysis (Hardoon et al., 2004), which ensures invariance to the matrices' bases' rotation. Additionally, because its correlation is single, its score is in [-1, 1].

It was shown that *QVEC* and *QVEC-CCA* are not limited to semantic evaluation, they can evaluate word vector content according to desired linguistic properties (Tsvetkov et al., 2016). *QVEC-CCA* outperforms other intrinsic evaluators on a range of *extrinsic* semantic and syntactic tasks. In addition, *QVEC-CCA* achieves a higher correlation with downstream tasks than existing word vector *intrinsic evaluation* methods based on word similarity.

2.1.5 Morphological similarity

In the morphologically rich embedding mentioned in Section 1.3.1 authors introduce a new metric specific for their WE (Cotterell and Schütze, 2015) which however can be interesting to compare different other WE methods. It is defined by the following formula

$$MorphoSim(w) = -\sum_{w' \in K_w} \min_{m_w, m_{w'}} (d_h(m_w, m_{w'})$$

, where:

- w is a word, K_w is a set of k nearest neighbors of w (k-NN)
- m_w is a vector representing morphological tags of w
- d_h is the Hamming distance between the vectors

.

	Morph-LBL	LBL	WORD2VEC
All Types	81.5%	22.1%	10.2%
No Tags	44.8%	15.3%	14.8%

Figure 8: MorphoSim results for Morphological LBL, classic LBL and Word2Vec. "No tags" shows experiments where the training data didn't include the words and their tags and "All types" represents no filtering of the test data (Cotterell and Schütze, 2015)

.

As shown in Figure 8, the morphological WE model greatly (and not surprisingly) outperforms classic LBL and Word2Vec. What is interesting in the results, is that for the authors' WE and classic LBL, there is a drop in the performance on unseen data (and in the case of Morphological LBL, it decreased almost twofold), however, Word2Vec performance has improved, potentially pointing towards Word2Vec strength of working with unseen data and extracting the morphological information from context similarity instead. Though the metric is only used to compare the authors' custom model with the not-enriched LBL and Word2Vec, it can be a powerful

tool to compare the morphologic richness of other WE models and be a sign of the importance of PoS information in the WE vector of these models.

2.2 Extrinsic Evaluation

Alongside *intrinsic* evaluators, *extrinsic* ones assess the quality of WE by submitting them to NLP downstream tasks.

2.2.1 PoS tagging

The *PoS tagging* task is to label each input token, such as an adjective, adverb, verb, noun, etc. Taking advantage of the convenience of labeled corpora, *PoS tagging* can be done by learning probability distributions through linguistic features or statistical machine learning (Wang et al., 2019).

2.2.2 Chunking

The purpose of *chunking* is to label segments of a sentence with syntactic constituents on the basis of *PoS tagging*. Each word is first assigned a label with its attributes, such as conjunctions or adjectives, and then the labeled words are syntactically grouped into related phrases (Wang et al., 2019).

2.2.3 NER

The task of *NER* is to recognize information units, such as names, which contain people's names, locations and organizations, and numeric expressions, which contain times and percentages. *NER* systems can use both language grammar-based techniques and statistical models. Grammar-based *NER* systems require considerable effort from experienced linguists. The *NER* system based on statistics requires a large amount of artificially labeled data for training, and its accuracy is higher than the former (Wang et al., 2019).

2.2.4 Sentiment Analysis

Sentiment analysis is a special text classification problem. It refers to labeling text fragments with binary or multivariate labels, corresponding to the positive or negative sentiment of the text. With the development of machine learning, many statistical and data-driven methods are used to deal with the task of *sentiment analysis* (Ravi and Ravi, 2015).

There is an assumption in the *extrinsic evaluation* of WE: the quality of WE has a fixed ranking, so higher-ranked embeddings will lead to better results in downstream tasks. However, by analyzing two downstream tasks of NER and *sentiment analysis*, researchers have shown that this assumption is not true, because it is proved

that using different evaluation criteria will produce different relative orderings of embeddings (Schnabel et al., 2015).

2.3 Correlation between Intrinsic-Extrinsic Evaluation

Most of the current studies on the *intrinsic* and *extrinsic* evaluation of WE are studied separately, and few studies specifically study the correlation between them. There is still some controversy about whether the *intrinsic evaluation* can predict downstream tasks. Earlier studies on the correlation between intrinsic-extrinsic evaluation mainly include (Chiu et al., 2016; Qiu et al., 2018). In (Wang et al., 2019), the authors pointed out that (Chiu et al., 2016) generating models only by changing the window size is not widely used in practical applications, and the results obtained may be biased. Furthermore, the limited experiments in (Qiu et al., 2018) are limited to Chinese only.

Therefore (Wang et al., 2019) attempts to provide a comprehensive study to avoid the pitfalls of previous work. This article studies the performance consistency of *extrinsic* and *intrinsic evaluators* by using Pearson correlation (p) analysis, aiming to help people choose appropriate WE models for specific tasks. The experiment is applied to 6 basic models, which are *Skip-gram*, *CBOW*, *GloVe*, *FastText*, *ngram2vec* and *Dict2vec*. There are also two *pre-trained GloVe* and *FastText* models, a total of 8 models. Next, variance normalization techniques were applied to these 8 models to generate another 8 models, resulting in 16 models in the end. The experimental results are shown in Figure 9.

In the experiment, it was shown that for the consistency of *intrinsic* evaluators:

- *word similarity*, *word analogy*, and *concept analogy*are three effective *intrinsic* raters
- different datasets affect the performance of the evaluators, and the results show that larger datasets produce better and more reliable results
- the performance of *intrinsic* evaluators varies widely across different downstream tasks. So the three *intrinsic* evaluators mentioned above should all be applied to test new WE models

For the consistency of *extrinsic* evaluators:

- since the performances of *PoS tagging*, *chunking* and *NER* depend on their ability to intrinsically order information, none of the *intrinsic* evaluators provides high correlation for *PoS tagging*, *chunking* and *NER*
- *sentiment analysis* has a stronger correlation with the properties of analogous evaluators because *sentiment analysis* focuses more on combinations of word meanings

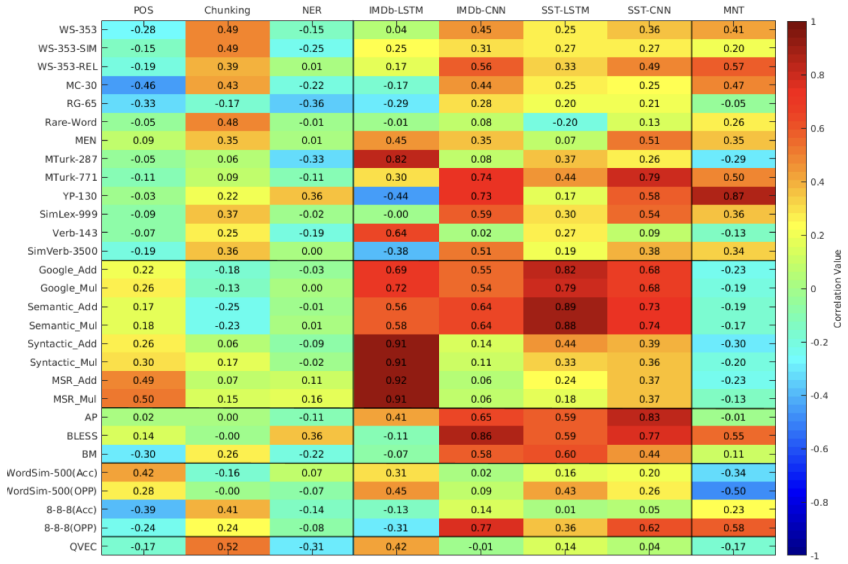

Figure 9: Pearson's correlation between intrinsic and extrinsic evaluator, where the x-axis shows extrinsic evaluators while the y-axis indicates intrinsic evaluators. The warm color indicates a positive correlation while the cool color indicates a negative correlation (Wang et al., 2019)

.

- Neural Machine Translation (NMT) has a stronger correlation with the properties of similarity evaluators, because NMT is a sentence-to-sentence transformation, and the mapping between word pairs is more helpful in translation tasks

3 Retrieval of linguistic information in Word Embeddings

While WE have become an inseparable part of most tasks of NLP, there has not been a lot of progress in understanding what data is encoded in the vectors themselves and how they can be interpreted. In this section, we will discuss the works that have made attempts in order to decipher the information encoded in WE vectors.

3.1 Analyzing Word Embeddings

The approaches for analyzing WE among researchers vary from trying to identify which particular parameter represents a specific semantic meaning (Jang and Myaeng, 2017), to exploring a WE as a whole (Gladkova and Drozd, 2016), to performing a Principal Component Analysis (PCA) of WE (Hollis and Westbury, 2016; Musil, 2019), to performing a cross-language analysis of properties encoded in WE (Bacon, 2020; Qian et al., 2016).

3.1.1 Analyzing a Singular Dimension

Some works focus on identifying which of WE parameters represent a specific semantic parameter (Jang and Myaeng, 2017). The authors use HyperLex (Vulić et al., 2017), a gold standard dataset that represents a set of categories and a set of concepts with ratings from 1 to 10 of how closely a particular concept is related to a particular category (for example, *apple* is highly likely to be a fruit and can be scored 9 in the $fruit$ category and *car* is unlikely to be a fruit and is scored 0 in the corresponding category). For their experiment, authors chose only to work with $food$, $fruit$ (a sub-category of $food$), *animal*, *bird* (sub-category of *animal*) and *instrument* (not closely related either with $food$ or *animal*), in total working with 139 words. For all of the corresponding terms the researchers obtained their WE using Word2Vec and found the components with a maximum average value for each category.

As shown in Figure 10, there are certain parameters (which authors have called **SigProps**) of WE that have the highest average value which is shared by category and their subcategories and there is no overlap with the unrelated category instruments which can be linked to the semantic similarity. The authors also performed a correlation analysis of typicality of terms and the values of the parameters in Figure 10 and found that some of the parameters are highly correlated with the typicality scores while others are not (Figure 11).

While the experiment size is very small and the categories used are very limited, this simple approach could be applied to analyzing WE for encoding morphological information as well, using PoS as categories and typicality scores calculated as a

Category	SIG-PROPS	
	Comp. ID	Avg.
instrument	c88	0.806
	c258	0.769
animal	**c154**	0.587
	c265	0.221
bird	**c154**	0.550
	c265	0.213
food	c207	0.298
	c233	0.269
fruit	c229	0.492
	c27	0.369
	c156	0.349
	c44	0.264
	c233	0.206

Figure 10: The parameters of WE with highest average (Jang and Myaeng, 2017)

.

SIG-PROPS	Correlation	Corr. rank
c229	0.743	1st
c233	0.540	4th
c27	0.516	5th
c44	0.474	7th
c156	-0.663	85th

Figure 11: Correlation between parameters and scores for typicality of concepts (Jang and Myaeng, 2017)

.

probability distribution of a particular word being a certain PoS. Applying SigProps analysis to morphological information could mean testing the parameters on a much bigger set of data.

Another critique of the experiment could be that the authors are only looking at maximum positive values, however, we don't know if a certain typicality may be encoded by the lowest average. Or a parameter closest to 0 on average? What if a set of parameters are responsible for encoding a concept typicality with a distribution that can't be caught by an average? In Section 3.1.2 we can have a look at these questions closer.

3.1.2 Analyzing Groups of Dimensions

Instead of looking at a certain particular dimension of WE, other researchers found that a big set of parameters of WE are changing together for semantically close terms (Gladkova and Drozd, 2016). For their experiment, the researchers chose 2 sets of words, a random set of terms (*emergency, bluff, buffet, horn, human, like, american, pretend, tongue, green*) and a set of semantically closely related terms (*cat, lion, tiger, leopard, cougar, cheetah, lynx, bobcat, panther, puma*). The researchers produced *GloVe* WE for these 2 sets and plotted the values of WE dimensions, highlighting the overlap as can be seen in Figure 12. In comparison to the SigProps experiment (Jang and Myaeng, 2017), we can clearly see that in the closely related terms (feline nouns) a whole range of WE parameters is maximized, while another range is minimized.

However, once again the experiment was performed only on small data which does not allow us to extrapolate the results to a larger scale due to the selected

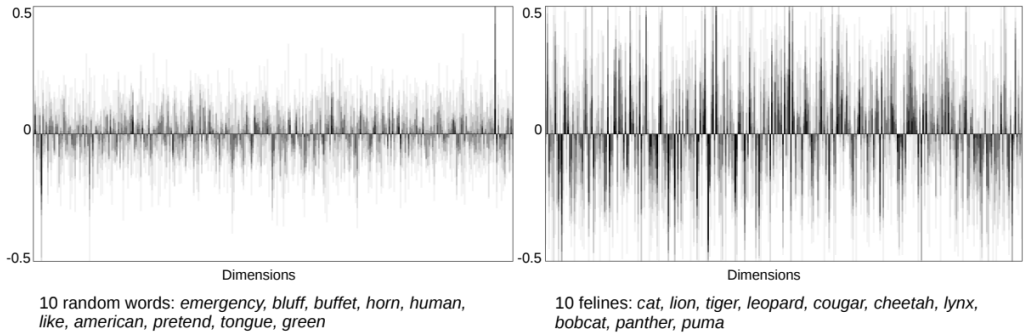

Figure 12: Heatmap of overlapping values of WE dimensions (Gladkova and Drozd, 2016)

.

test datasets. However, the frame of it allows us to apply it for the task of morphological analysis as well, to validate if the same type of parameter maximization/minimization is only characteristic for semantically related datasets or whether it's applicable for morphologically related sets.

In another work, the researchers set 3 lexical variables (*log frequency* (from HAL corpus), *orthographic neighborhood size*, *word length*) to test WE for and 5 semantic ones (*valence, arousal, dominance, concreteness, age of acquisition*) (Hollis and Westbury, 2016). To achieve this, they created Skip-gram embeddings using Google News corpus. During their study, the authors found a strong correlation of valence judgments with 79% (208) of dimensions of the WE they obtained. Similarly, they found that each of the other variables they have set had a strong correlation with above 50% of all of WE dimensions. It could be argued that such results are a consequence of very broad variables set by the researchers, interpreting *concretness*, for example, can be pretty confusing even for human analyzers. This point of fuzziness and interpretability of WE will be further discussed in Section 3.2.

3.1.3 Principal Component Analysis of Word Embeddings

As we could see in Section 3.1.2, multiple parameters of WE are maximized or minimized when we're looking into embeddings of related word sets. Since there are many components at play, it is reasonable to perform a PCA of WE.

In the same experiment mentioned in Section 3.1.2, the researchers have attempted to interpret principal components (PC) as their originally set variables (Hollis and Westbury, 2016). They have found that there is a strong correlation of PC1 with *word frequency*, PC2 with *concreteness*, PC5 with *valence*, and PC7 with *dominance*. However, they have also discovered that it required 254 PCs to cover 95% of the variance of the sample set and 23 PCs were strongly correlated with

valence. Furthermore, while analyzing the words loaded highly on PC1 (which they connected) with *word frequency*, they found words like *implement, evaluate, finalize, strengthen* and on the lower side of PC1 they found *herub, puss, wienie, hussy, and senorita* which at a glance represent not the most popular or unpopular words but instead formal vs informal terms. Thus, even when having a strong correlation with a feature, it is still hard to interpret a PC as a particular feature.

It could be argued that the problem of interpretability of PCs lies solely in fuzzy variables that the researchers tried to match with them, however, in another paper, it was attempted to correlate PCs with PoS which is less likely subject to confusion (Musil, 2019). In this work, it was shown that for Word2Vec embeddings trained on a Czech corpus (see Figure 13) a word being a noun or verb is correlated with 3 PCs of the WE. What is interesting in this chart is that there is a strong negative correlation between nouns and verbs for all of these 3 components.

To summarize, analyzing PCs of WE we can not distinctly say that one of PCs is responsible for a particular feature, there is fuzziness for both semantic and morphological features. Therefore, we propose the following experiment:

1. Perform PCA and correlate with PoS features to avoid additional fuzziness of interpretability
2. For highly correlated PCs, validate the linear combination of the original features (dimensions of WE) to see if there are one or more dimensions that are shared by the PCs and could be connected to the PoS information
3. Form 3 different WE matrices for different PoS: nouns, verbs and adjectives, and adverbs and run PCA on them and correlate with morphological features (tense, singular/plural, feminine/masculine, etc.)
4. From the correlated PCs attempt to retrace it back to the original dimensions

To elaborate on Steps 4 and 5: from the previous works, we understand that there is high variance and a lot of dimensions that PCs represent. Since we do not pursue the task of PoS-clustering based on WE, we can create less variance by isolating the PoS and looking at them separately. It was previously shown as well that broader categories and features seem to be correlated with more PCs, to address this issue we will try to make the features we are looking for more granular and defined.

3.2 Possibility of Interpretation of Word Embeddings

One of the big questions that still remains: is a human interpretation of WE even possible? As previous examples of analysis of WE embeddings have shown, a lot relies on gold standards and human evaluation. Using a typicality score (Jang and Myaeng, 2017) depends on manual scoring of terms but is it feasible and reliable?

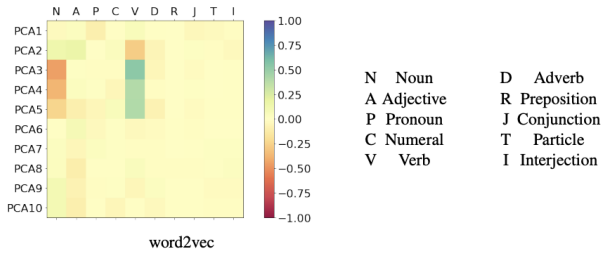

Figure 13: Correlation of PC and PoS for word2vec, Czech language (Musil, 2019)

.

Do participants of the evaluation understand concepts like *typicality* in the same way? With *apple* now standing not only for fruit but an IT company, should there be any effect on the fruit typicality score? There is even less certainty if evaluators have the same understanding of abstract concepts like *concreteness* when they are evaluating it. Even with the PoS classification being more straightforward, there is still room for discussion and disagreement between the annotators (see Figure 14).

Does this mean it is impossible for us to find out how any information is encoded in WE? We will attempt to answer this question in our research, our main approach is to make the features we analyze very narrow and allowing little space for interpretation. For the case of French, we will work with nouns, verbs and adjectives where there is little chance for confusion between annotators. Moreover, we look for very granular features in the WE: gender, plurality, tense, etc. In our research, we will answer if this approach can interpret WE with any more clarity than previous attempts.

Figure 14: Different annotation of the word *quelque* in the same corpus, French GSD [5]

[5]https://universaldependencies.org/treebanks/fr_gsd/index.html

4 Conclusion and Future Work

WE are an inseparable part of many NLP research fields and it has been shown that the quality of WE and intrinsic task performance is correlated with the performance of the NLP tasks. However, we still lack a definitive understanding of how and where a piece of information about a word and its context is encoded in a WE.

In our project, we will generate WE in the French language using Word2Vec (*CBOW* and *Skip-gram*) and *GloVe* algorithms. Next, we aim to perform an intrinsic evaluation using the intrinsic evaluators mentioned above in Section 2.1. Moreover, we will attempt to extract morphological information from the vectors by analyzing the dimensions of WE both individually and as linear combinations using PCA.

Previous works that attempted extracting linguistic features from WE have shown that there is a lot of uncertainty which we will try to address by narrowing the scope of the research toward looking for simple morphological information which doesn't cause additional misinterpretation. Additionally, we will attempt to answer the question of whether the interpretability of WE is possible using our experiment results.

References

Nourredine Aliane. 2019. *Evaluation des représentations vectorielles de mots.* Ph.D. thesis, Paris 8.

Mikel Artetxe, Gorka Labaka, Iñigo Lopez-Gazpio, and Eneko Agirre. 2018. Uncovering divergent linguistic information in word embeddings with lessons for intrinsic and extrinsic evaluation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 282–291, Brussels, Belgium. Association for Computational Linguistics.

Geoffrey I Bacon. 2020. *Evaluating linguistic knowledge in neural networks.* Ph.D. thesis, UC Berkley.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. 2016. A critique of word similarity as a method for evaluating distributional semantic models. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 7–12, Berlin, Germany. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, page 1137–1155.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292, Denver, Colorado. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 36–42, Berlin, Germany. Association for Computational Linguistics.

David Roi Hardoon, Sándor Szedmák, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. In *Neural Computation*, volume 16, pages 2639–2664.

Zellig S. Harris. 1954. Distributional structure. In *WORD*, volume 10, pages 146–162. Routledge.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. In *Computational Linguistics*, volume 41, pages 665–695, Cambridge, MA. MIT Press.

Geoff Hollis and Chris Westbury. 2016. The principals of meaning: Extracting semantic dimensions from co-occurrence models of semantics. In *Psychonomic bulletin & review*, volume 23, pages 1744–1756. Springer.

Kyoung-Rok Jang and Sung-Hyon Myaeng. 2017. Elucidating conceptual properties from word embeddings. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 91–95, Valencia, Spain. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010a. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048. ISCA.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *ArXiv*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013c. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013d. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010b. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2010, pages 1045–1048. International Speech Communication Association.

Tomá Musil. 2019. Examining structure of word embeddings with pca. In *International Conference on Text, Speech and Dialogue*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Investigating language universal and specific properties in word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. 2018. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 209–221, Cham. Springer International Publishing.

Kumar Ravi and Vadlamani Ravi. 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. In *Knowledge Based Systems*, volume 89, pages 14–46.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. 2nd International Conference on Learning Representations.

Yulia Tsvetkov, Manaal Faruqui, and Chris Dyer. 2016. Correlation-based intrinsic evaluation of word vector representations. In *Proceedings of the 1st Workshop*

on Evaluating Vector-Space Representations for NLP, pages 111–115, Berlin, Germany. Association for Computational Linguistics.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. In *Computational Linguistics*, volume 43, pages 781–835, Cambridge, MA.

Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C.-C. Jay Kuo. 2019. Evaluating word embedding models: methods and experimental results. volume 8. Cambridge University Press.