A STUDY ON PROBABILISTIC WORD EMBEDDING: REPRESENTING WORDS AS PROBABILITY DISTRIBUTIONS

MS(QMS) Dissertation

Krishnendu Pain
Advisor: GSR Murthy

INDIAN STATISTICAL INSTITUTE

CERTIFICATE

This is to certify that Mr. Krishnendu Pain (Roll No. MQMS2005), a student of II-year MS(QMS) Course (Batch: 2020-22) has undertaken the dissertation work titled “A Study on Probabilistic Word Embedding: Representing Words as Probability Distributions” under my guidance.

This report is the bonafide record of the work done by him/her during 10th January 2022 to 5 March 2022 and submitted to the institute, in partial fulfilment of the course requirements.

Date: February/March 2022

Guide Signature

CONTENTS

A Study on Probabilistic Word Embedding: Representing Words as Probability Distributions

Krishnendu Pain

Advisor: GSR Murthy

Abstract

A word embedding is a learned representation for text where words that have the same meaning have a similar representation. In current practice of Natural Language Processing, representing word as point vectors is most popular method. But there is another methodology to represent word, which has opened a new path in NLP practices, and that is representing word by a probability distribution, specifically, the Gaussian distribution. In this study, we review the methodology of representing word by a multimodal Gaussian distribution and provide a simple overview and explanation of the procedure.

Keywords: Word Embedding, Natural Language Processing, Gaussian Distribution, Multimodal word distributions.

1.0 Introduction

Representation of word is a fundamental task in language modelling. There are many techniques available for the task, starting from presenting every word as a binary one-hot vector corresponding to a dictionary. But modern approaches which learn to map words with similar meanings to nearby point vectors in a vector space [Mikolov et al., 2013] from large datasets, are superior to those basic methods. These modern methodologies of word vectorization have become a go-to approach in NLP tasks.

There is an alternative proposal of word embedding provided by Vilnis and McCallum (2014), where words are represented by a whole probability distribution, specifically, Gaussian distribution instead of a point vector, and learn its mean and covariance matrix from data. This approach generalizes any deterministic point embedding, which can be fully captured by the mean vector of the Gaussian distribution. Moreover, the full distribution provides much richer information than point estimates for characterizing words, representing probability mass and uncertainty across a set of semantics.

Since a Gaussian distribution can have only one mode, the learned uncertainty in this representation can be overly diffuse for words with multiple distinct meanings. Athiwaratkun and Wilson (2017) proposed an improved method where they tried to represent each word with an expressive multimodal distribution for multiple distinct meanings, entailment, heavy tailed uncertainty and enhanced interpretability. Particularly, they model each word with a mixture of Gaussians, by learning all the parameters of this mixture using a maximum margin energy-based ranking objective [Joachims, 2002; Vilnis et al., 2014], where the energy function describes the affinity between a pair of words. For analytic tractability with Gaussian mixtures, the inner product

between probability distributions in a Hilbert space, known as the expected likelihood kernel [Jebara et al., 2004] is used as energy function.

2.0 Related Work and Literature Review

In current NLP practices, word2vec is arguably the most popular word embedding method. This method uses continuous bag of words and skip-gram models, in conjunction with negative sampling for efficient conditional probability estimation [Mikolov et al., 2013]. The continuous bag of words method or CBOW model tries to understand the context of the words and takes this as input, and then it tries to predict words that are contextually accurate. The skip-gram model learns by predicting the surrounding words given a current word, within a certain range before and after the current word.

A different approach to learning word embeddings is through factorization of word cooccurrence matrices such as Glove embeddings [Pennington et al., 2014]. The matrix factorization approach has been shown to have an implicit connection with skip-gram and negative sampling [Levy et al., 2014].

Vilnis and McCallum (2014) proposed a Gaussian distribution to model each word. This method is more expressive than typical point embeddings, with the ability to represent concepts such as entailment, by having the distribution for one word (e.g., 'music') encompass the distribution for sets of related words ('jazz' and 'pop'). But with a unimodal distribution, this approach cannot capture multiple distinct meanings.

Athiwaratkun and Wilson (2017) proposed a probabilistic word embedding method that can capture multiple meanings. They have used a Gaussian mixture model which allows for a highly expressive distributions over words. In this method, scalability and analytical tractability with an expected likelihood kernel energy function for training is retained. This model and training procedure harmonize to learn descriptive representations of words with superior performance on several benchmarks.

3.0 Methodology

We will simplify and describe the methodology for multimodal Gaussian embedding in this section. The Gaussian Mixture (GM) model for word representation and a training method to learn the parameters of the mixture will be introduced, along with an energy function that compliments the model by retaining analytic tractability.

3.1 Word Representation

Each word w in a dictionary is represented as a Gaussian mixture with K components. The distribution of w, f_w, is given by the density

$$f_w(\vec{x}) = \sum_{i=1}^{K} p_{w,i} N\left[\vec{x};\ \vec{\mu}_{w,i}, \Sigma_{w,i}\right] \qquad (1)$$
$$= \sum_{i=1}^{K} \frac{p_{w,i}}{\sqrt{2\pi|\Sigma_{w,i}|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_{w,i})^T \Sigma_{w,i}^{-1} (\vec{x}-\vec{\mu}_{w,i})}$$

where $\sum_{i=1}^{K} p_{w,i} = 1$.

The mean vectors $\vec{\mu}_{w,i}$ represent the location of the i^{th} component of word w. $p_{w,i}$ represents the component probability (mixture weight) and $\Sigma_{w,i}$ is the component covariance matrix, containing uncertainty information. The goal of this method is to learn all of the model parameters $\vec{\mu}_{w,i}, p_{w,i}, \Sigma_{w,i}$ from a corpus of natural sentences to extract semantic information of words. Each Gaussian component's mean vector of word w can represent one of the word's distinct meanings. As an example, consider the word 'rock'. One component of the word should represent the meaning related to 'stone', whereas other component should represent the meaning related to 'music.'

3.2 Training Procedure

The training objective for learning is $\theta = \{\vec{\mu}_{w,i}, p_{w,i}, \Sigma_{w,i}\}$. This training procedure is similar to the continuous skip-gram model [Mikolov et al., 2013], where word embeddings are trained to maximize the probability of observing a word given another word. This process follows the distributional hypothesis that words occurring in contexts tend to be semantically related. For example, the words 'pop' and 'music' tend to occur nearer to each other than the words 'pop' and 'tiger'; hence, 'pop' and 'music' have higher correlation compared to the other pair of words.

3.3 Energy-based Max-Margin Objective

In this objective, every sample consists of two pairs of words, (w, c) and (w, c'). w is the word sampled from a sentence in the corpus and c is a nearby word within a context window l. For example, consider the sentence 'I am a good boy'; the word $w =$ 'good' has context words 'I', 'am', 'a', 'boy'. A negative context word c' is a word obtained from random sampling. For example, for the previous sentence, the word 'mathematics' is a negative context word, i.e., it has no correspondence with the word $w =$ 'good'.

The objective is to maximize the energy between words that occur near each other, w and c, and minimize the energy between w and its negative context c'. The energy function is a measure of similarity between distributions.

A max-margin ranking objective [Joachims, 2002] is used here. It is also used for Gaussian embeddings [Vilnis et al., 2014], which pushes the similarity of a word and its positive context higher than of its negative context by a margin m:

$$L_\theta(w, c, c') = \max(0, m - \log E_\theta(w, c) + \log E_\theta(w, c')$$

The minimization method for this objective can be done by mini-batch stochastic gradient descent with respect to the parameters $\theta = \{\vec{\mu}_{w,i}, p_{w,i}, \Sigma_{w,i}\}$ of the multimodal embedding in Eq. (1).

3.4 Energy Function

For popular practices of word vectorization, a usual choice for energy function is a dot product between two vectors. But here the words are represented as probability distributions instead of point vectors, so a measure is required so that it can reflect both similarity and uncertainty.

In this study, the expected likelihood kernel, which is a generalization of an inner product between vectors to an inner product between distributions [Jebara et al., 2004], i.e.,

$$E(f,g) = \int f(x)g(x)\,dx = \langle f,g\rangle_{L_2}$$

where $\langle\cdot,\cdot\rangle_{L_2}$ denotes the inner product in Hilbert space L_2. This form of energy is chosen as it can be evaluated in a closed form given the choice of embedding in Eq. (1).

Consider two words w_f, w_g, and Gaussian mixtures representing them are f and g respectively, where,

$$f(x) = \sum_{i=1}^{K} p_i N\left[\vec{x};\ \vec{\mu}_{w_f,i}, \Sigma_{w_f,i}\right] \quad \text{and} \quad g(x) = \sum_{i=1}^{K} q_i N\left[\vec{x};\ \vec{\mu}_{w_g,i}, \Sigma_{w_g,i}\right]$$

and $\sum_{i=1}^{K} p_i = 1,\ \sum_{i=1}^{K} q_i = 1$

The log energy is,

$$\log E_\theta(f,g) = \log \sum_{j=1}^{K}\sum_{i=1}^{K} p_i q_j e^{\xi_{i,j}} \qquad (2)$$

where,

$$\begin{aligned}\xi_{i,j} &= \log N\left(0;\ \vec{\mu}_{w_f,i} - \vec{\mu}_{w_g,j}, \Sigma_{w_f,i} + \Sigma_{w_g,j}\right)\\ &= -\frac{1}{2}\log\det\left(\Sigma_{w_f,i} + \Sigma_{w_g,j}\right) - \frac{D}{2}\log(2\pi)\\ &\quad - \frac{1}{2}\left(\vec{\mu}_{w_f,i} - \vec{\mu}_{w_g,j}\right)^T\left(\Sigma_{w_f,i} + \Sigma_{w_g,j}\right)^{-1}\left(\vec{\mu}_{w_f,i} - \vec{\mu}_{w_g,j}\right) \qquad (3)\end{aligned}$$

The term $\xi_{i,j}$ is called partial (log) energy. This term expresses the similarity of between the i^{th} meaning of word w_f and the j^{th} meaning of word w_g. The energy in Eq. (2) is the sum of all possible pairs of partial energies, weighted accordingly by the mixture probabilities p_i and q_j.

The term $-\left(\vec{\mu}_{w_f,i} - \vec{\mu}_{w_g,j}\right)^T\left(\Sigma_{w_f,i} + \Sigma_{w_g,j}\right)^{-1}\left(\vec{\mu}_{w_f,i} - \vec{\mu}_{w_g,j}\right)$ in $\xi_{i,j}$ explains the difference in mean vectors of semantic pair (w_f, i) and (w_g, j). If the covariance for both pairs are low, this term has more importance relative to other terms due to the inverse covariance scaling. The loss function L_θ attains a low value when $E_\theta(w,c)$ is relatively high. We can obtain high values of $E_\theta(w,c)$ when the component means across different words $\vec{\mu}_{w_f,i}$ and $\vec{\mu}_{w_g,j}$ have similar point representation; this can also be done by large values of $\Sigma_{w_f,i}$ and $\Sigma_{w_g,j}$, breaking the importance of the mean vector difference. The term $-\log\det\left(\Sigma_{w_f,i} + \Sigma_{w_g,j}\right)$ serves as a regularizer that prevents the covariances from being pushed too high at the expense of learning a good mean embedding.

4.0 Application

The proposed mixture model is trained on a concatenation of two datasets: UKWAC (2.5 billion tokens) and Wackypedia (1 billion tokens) [Baroni et al., 2009], to learn the parameters. After training, learned parameters $\{\vec{\mu}_{w,i}, p_{w,i}, \Sigma_{w,i}\}_{i=1}^{K}$ for each word w is obtained. The mean vector $\vec{\mu}_{w,i}$ is treated as the embedding of the i^{th} mixture component with the covariance matrix $\Sigma_{w,i}$ representing its subtlety and uncertainty. This model is named as Word to Gaussian Mixture (w2gm).

4.1 Hyperparameters

The experiment is done with $K = 2$ components for the w2gm model, but this also can be done with $K = 3$. The reason behind this choice is most of the polysemous words have two meanings, there are very small number of words with three or more meanings. For this experiment the spherical covariance is considered, as for diagonal or spherical covariances, the matrix can be computed very efficiently since the matrix inversion would require $\mathrm{O}(d)$ computation instead of $\mathrm{O}(d^3)$ for a full matrix.

4.2 Similarity Measures

The word embeddings contain multiple vectors and uncertainty parameters per word. The following measures are used to measure similarity –

4.2.1 Expected Likelihood Kernel

This is a natural choice for similarity score, as this is an inner product between distributions. This metric measures the uncertainty from the covariance matrices in addition to the similarity between the mean vectors.

4.2.2 Maximum Cosine Similarity

This metric measures the maximum similarity of mean vectors among all pairs of mixture components between distributions f and g, i.e., $d(f, g) = \max_{i,j=1,\ldots,K} \frac{\langle \mu_{f,i}, \mu_{g,j} \rangle}{\|\mu_{f,i}\| \cdot \|\mu_{g,j}\|}$, which corresponds to matching the meanings of f and g that are the most similar.

4.2.3 Minimum Euclidean Distance

The training objective in Eq. (3) directly involves the Euclidean distance, as opposed to dot product of vectors such as in word2vec. So, the Euclidean metric is also considered: $d(f, g) = \min_{i,j=1,\ldots,K} \|\mu_{f,i} - \mu_{g,j}\|$.

4.3 Qualitative Evaluation

Examples of polysemous words and their nearest neighbors in the embedding space is shown in Table 1 to demonstrate the ability of the embedding method.

Word	Co.	Nearest Neighbors
rock	0	basalt:1, boulder:1, boulders:0, stalagmites:0, stalactites:0, rocks:1, sand:0, quartzite:1, bedrock:0
rock	1	rock/:1, ska:0, funk:1, pop-rock:1, punk:1, indie-rock:0, band:0, indie:0, pop:1
bank	0	banks:1, mouth:1, river:1, River:0, confluence:0, waterway:1, downstream:1, upstream:0, dammed:0
bank	1	banks:0, banking:1, banker:0, Banks:1, bankas:1, Citibank:1, Interbank:1, Bankers:0, transactions:1
Apple	0	Strawberry:0, Tomato:1, Raspberry:1, Blackberry:1, Apples:0, Pineapple:1, Grape:1, Lemon:0
Apple	1	Macintosh:1, Mac:1, OS:1, Amiga:0, Compaq:0, Atari:1, PC:1, Windows:0, iMac:0
star	0	stars:0, Quaid:0, starlet:0, Dafoe:0, Stallone:0, Geena:0, Niro:0, Zeta-Jones:1, superstar:0
star	1	stars:1, brightest:0, Milky:0, constellation:1, stellar:0, nebula:1, galactic:1, supernova:1, Ophiuchus:1
cell	0	cellular:0, Nextel:0, 2-line:0, Sprint:0, phones.:1, pda:1, handset:0, handsets:1, pushbuttons:0
cell	1	cytoplasm:0, vesicle:0, cytoplasmic:1, macrophages:0, secreted:1, membrane:0, mitotic:0, endocytosis:1
left	0	After:1, back:0, finally:1, eventually:0, broke:0, joined:1, returned:1, after:1, soon:0
left	1	right-hand:0, hand:0, right:0, left-hand:0, lefthand:0, arrow:0, turn:0, righthand:0, Left:0

Word	Nearest Neighbors
rock	band, bands, Rock, indie, Stones, breakbeat, punk, electronica, funk

bank	banks, banking, trader, trading, Bank, capital, Banco, bankers, cash
Apple	Macintosh, Microsoft, Windows, Macs, Lite, Intel, Desktop, WordPerfect, Mac
star	stars, stellar, brightest, Stars, Galaxy, Stardust, eclipsing, stars., Star
cell	cells, DNA, cellular, cytoplasm, membrane, peptide, macrophages, suppressor, vesicles
left	leaving, turned, back, then, After, after, immediately, broke, end

Table 1: Nearest neighbors based on cosine similarity between the mean vectors of Gaussian components for Gaussian mixture embedding (for $K = 2$) and Gaussian embedding. The notation $w: i$ denotes the i^{th} mixture component of the word w

Take the example for the word 'rock'. It has two meanings 'stone' and a form of music ('rock music'). This word should have each of its meanings represented by a distinct Gaussian component. From the result in Table 1, we can see that the 0^{th} component of 'rock' being related to the words 'basalt', 'boulders' and the 1^{st} component being related to 'funk', 'indie', 'hip-hop'.
Similarly, consider the word 'bank'. Its 0^{th} component represents the river bank (as the related words are 'river', 'waterway', 'downstream') and the 1^{st} component represents the financial bank (as the related words are 'transactions', 'Citibank', 'banking').

By contrast, in Table 1 (bottom), we can see that for Gaussian embeddings with one mixture component, nearest neighbors of polysemous words are predominantly related to a single meaning. For example, the words related to 'rock' are mostly regarding rock music. Similarly, for the word 'bank', related words are mostly regarding financial bank.

In Fig. 1, we can see that for the word 'rock', the top image represents the Gaussian mixture embedding with $K = 2$ components, and the bottom image represents the Gaussian embedding (same as Gaussian mixture with $K = 1$).

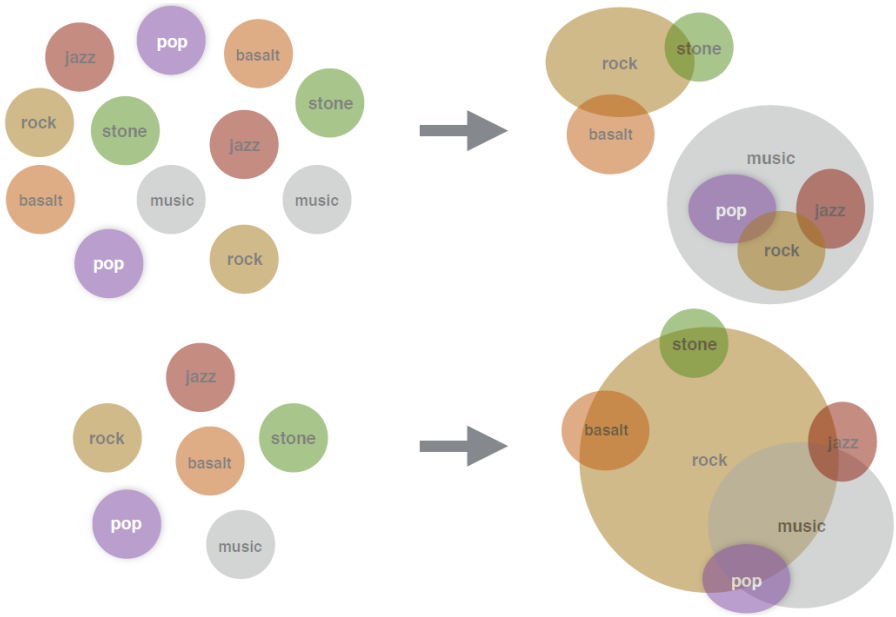

Figure 1: Visualization of word embedding (Top: Gaussian Mixture embedding with $K = 2$ and Bottom: Gaussian embedding)

For the cases where a word only has a single meaning, the mixture components can be very close; for example, consider the word 'stone'. One component of this is close to

('stones', 'stonework', 'slab') and the other is close to ('carving', 'relic', 'excavated'), which reflects subtle variations in meanings.

4.4 Word Similarity

The evaluation of the embeddings is done on several standard word similarity datasets, namely, SimLex [Hill et al., 2014], WS or WordSim-353, WS-S (similarity), WS-R (relatedness) [Finkelstein et al., 2002], MEN [Bruni et al., 2014], MC [Miller et al., 1991], RG [Rubenstein et al., 1965] and YP (Yang et al., 2006]. Each dataset contains a list of word pairs with a human score of how related or similar the two words are.

The Spearman correlation [Spearman, 1904] between the labels and our scores generated by the embeddings is calculated. It is a rank-based correlation measure that assesses how well the scores describe the true labels.

The results of the Gaussian mixture model are compared with the performance of word2vec and the original Gaussian embedding [Vilnis et al., 2014] in Table 2.

Dataset	word2vec	w2g	w2gm/mc	w2gm/el	w2gm/me
SL	29.39	**32.23**	23.31	26.02	27.59
WS	59.89	65.49	**73.47**	62.85	66.39
WS-S	69.86	76.15	**76.73**	70.08	73.3
WS-R	53.03	58.96	**71.75**	57.98	60.13
MEN	70.27	71.31	**73.55**	68.5	67.7
MC	63.96	70.41	79.08	76.75	**80.33**
RG	70.01	71	**74.51**	71.55	73.52
YP	39.34	41.5	**45.07**	39.18	38.58

Table 2: Spearman correlation for word similarity datasets; w2g and w2gm denote Gaussian embedding and Gaussian mixture embedding ($K = 2$); mc, el, me denote maximum cosine similarity, expected likelihood kernel and minimum Euclidean distance respectively.

From the results in Table 2, we can see that for most of the datasets, w2gm model has performed the best, with the maximum cosine similarity as the similarity measure. This model has also outperformed the most popular method word2vec for most of the datasets, namely, SL, WS, WS-S, WS-R, MEN, RG and YP. The minimum Euclidean distance is a better metric for the MC dataset.

5.0 Conclusion

In this study, a word embedding model which represents words with multimodal distributions formed from the Gaussian mixture is discussed. To learn the parameters of each mixture, an analytic energy function for combination with a maximum margin objective is proposed. The embeddings capture different semantics of polysemous words, and also perform well on word similarity benchmarks.

Probabilistic word embedding is a relatively new idea, which has the potential to model language in a more expressive way, and to make advancement in the state of the art. Multimodal word distributions has shown that the shape of a word distribution can express much more semantic information than any point representation.

We tried to come with a new idea of representing words as discrete distribution over a dictionary $d = [w_1, w_2, \ldots\ldots, w_N]$. Let, N be the dimension (number of elements in d), and $P = (p_{ij})$ be the $N \times N$ stochastic matrix where p_{ij} is the probability that word w_j is associated with the word w_i. We can use the energy functions mentioned before to represent the distances between words. The challenge in this procedure will be to handle the huge stochastic matrix which is a sparse matrix and do further operations.

This methodology has the potential to open the doors of a new path of applications in language modelling, building new types of supervised language models constructed to more fully leverage the rich information provided by word distribution.

References

1) Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean [2013]. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.
2) Luke Vilnis and Andrew McCallum [2014]. Word representations via Gaussian embedding. *CoRR* abs/1412.6623.
3) Ben Athiwaratkun and Andrew Gordon Wilson [2017]. Multimodal word distributions. *ACL*.
4) Thorsten Joachims. 2002. Optimizing search engines using clickthrough data [2002]. *In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada*. pages 133–142.
5) Tony Jebara, Risi Kondor, and Andrew Howard [2004]. Probability product kernels. *Journal of Machine Learning Research* 5:819–844.
6) 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 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543.
7) Omer Levy and Yoav Goldberg [2014]. Neural word embedding as implicit matrix factorization. *In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada*. Pages 2177–2185.
8) Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta [2009]. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation* 43(3):209–226.
9) Felix Hill, Roi Reichart, and Anna Korhonen [2014]. Simlex- 999: Evaluating semantic models with (genuine) similarity estimation. *CoRR* abs/1408.3456.
10) Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin [2002]. Placing search in context: the concept revisited. ACM Trans. *Inf. Syst.* 20(1):116–131.
11) Elia Bruni, Nam Khanh Tran, and Marco Baroni [2014]. Multimodal distributional semantics. J. Artif. *Int. Res.* 49(1):1–47.
12) George A. Miller and Walter G. Charles [1991]. Contextual Correlates of Semantic Similarity. *Language & Cognitive Processes* 6(1):1–28.
13) Herbert Rubenstein and John B. Goodenough [1965]. Contextual correlates of synonymy. Commun. *ACM* 8(10):627–633.
14) Dongqiang Yang and David M. W. Powers [2006]. Verb similarity on the taxonomy of wordnet. *In In the 3rd International WordNet Conference (GWC-06), Jeju Island, Korea*.
15) C. Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology* 15:88–103.