Programmatic Link Grammar Induction for Unsupervised Language Learning

Alex Glushchenko[1], Andres Suarez[1,2], Anton Kolonin[1], Ben Goertzel[1,2], Oleg Baskov[1]

[1]SingularityNET Foundation, Amsterdam, Netherlands

[2]Hanson Robotics, Hong Kong, China

{anton, ben}@singularitynet.io

Abstract. Although natural (i.e. human) languages do not seem to follow a strictly formal grammar, their structure analysis and generation can be approximated by one. Having such a grammar is an important tool for programmatic language understanding. Due to the huge number of natural languages and their variations, processing tools that rely on human intervention are available only for the most popular ones. We explore the problem of unsupervisedly inducing a formal grammar for any language, using the Link Grammar paradigm, from unannotated parses also obtained without supervision from an input corpus. The details of our state-of-the-art grammar induction technology and its evaluation techniques are described, as well as preliminary results of its application on both synthetic and real world text-corpora.

Keywords: categorization, clustering, computational linguistics, dimensionality reduction, formal grammar, grammar induction, natural language processing, unsupervised learning, vector space

1 Introduction

This work is grounded on the premise that the grammar of any language may be derived (at least to some extent) in an unsupervised way from statistical evidence of word co-occurrences observed in large unannotated corpora [1]. Following this idea, Vepstas and Goertzel [2] proposed to use such learned grammar for programmatic unlabeled dependency text parsing and part-of-speech tagging of raw text, for further extraction of semantics. The Link Grammar (LG) formalism [3] is proposed to represent the learned grammars, while parses are built by a maximum spanning tree (MST) algorithm [2].

In earlier work [4] we have described the implementation of the software framework capable to solve the described unsupervised language learning problem (to some extent) for synthetic English corpora, and approach the solution for real-world Eng-

lish corpora. The major components of our research pipeline (see Fig. 1) are: text tokenization, word-sense disambiguation (WSD), parsing, and grammar learning, with subsequent indirect evaluation of the produced grammar (by producing LG parses for a test corpus using the synthetic grammar and comparing them against expected parses).

Although text tokenization is a problem that can be attacked in an unsupervised manner [5], our current work has not attempted this seriously; for now, we rely on a rule-based English tokenizer. The WSD part of pipeline has shown promising results in earlier works [4,6], providing noticeable improvement in the quality of the learned grammars and it is not discussed herein.

The MST-parser has proved to be a critical component of our pipeline, as it provides input to the grammar induction process. Ongoing development in this area is worth separate discussion, but its importance is confirmed by the findings presented below.

This paper focuses on the part of the pipeline responsible for induction of a Link Grammar dictionary from input parses, on the process for evaluation of such grammars, as well as on the results obtained from our research efforts.

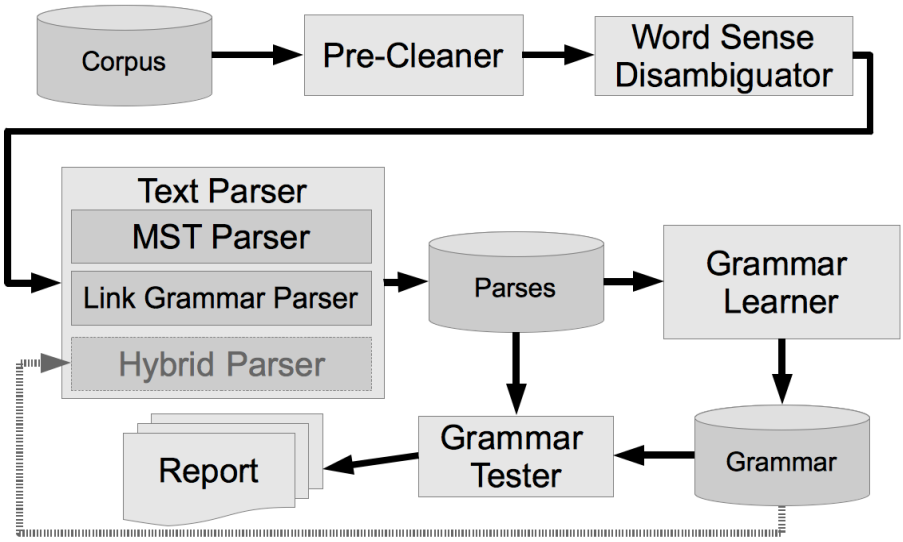

Fig. 1. Overall architecture of the unsupervised language learning pipeline, composed of a Pre-Cleaner responsible for tokenization, Text Parser (using either MST-Parser, or Link Grammar Parser or Hybrid Parser combining results of the previous two), Grammar Learner which induces a grammar from parses, and Grammar Tester that evaluates the learned grammar.

The fundamental importance of this research is based on the assumption that understanding natural human language acquisition is one of the keys to decipher the nature of human intelligence [7] and unlock the path to artificial general intelligence (AGI) [8]. Unlike other approaches to unsupervised language acquisition [9], our framework creates a language model that, in contrast to a neural network “black box”,

consists of a human-comprehensible formal grammar contained in a LG dictionary file. Such file lists grammar rules that can be further reviewed, edited and extended by human computational linguists, or used by the Link Parser software (https://github.com/opencog/link-grammar) to parse previously unseen text in the target language.

From a practical standpoint, the goal of the unsupervised language learning (ULL) project is to automate the process of building, or extending, formal grammars of human languages. These grammars could then be applied on the comprehension and production of text and speech in computer software, and artificial intelligence applications involving natural language processing.

2 Grammar Induction Architecture and Implementation

Our proposed method for grammar induction, part of the open-source OpenCog Unsupervised Language Learning (ULL) project, is implemented as its Grammar Learner (GL) component and is represented on Fig. 2 (code can be found at https://github.com/singnet/language-learning). This section dissects the steps necessary for this process.

The Grammar Learner component takes as input a set of dependency parses with undirected unlabeled links, which are used to create a word-vector space. Inspired by representations using a Shifted Positive Pointwise Mutual Information word-context matrix [11], the created word space is described by a sparse matrix M in which each row i corresponds to a word, each column j to a context in which the word appeared, and each matrix entry Mij corresponds to some association measure between the word and the context. From a given input parse, we extract each word's connectors as those context-words linked to it, as well as a label "-" if the context-word appears to the left of the reference word in the sentence, or a label "+" otherwise. A connector-set for a word (also called a "disjunct" [4]) is composed of all the connectors it has in a given parse tree. We then build the word-vector space matrix using either connectors-sets (for smaller corpora) or plain connectors (for the larger "Gutenberg Children Books" dataset) as the words contexts.

A variety of interaction metrics can be used as association measures: mutual information [12] and co-occurrence frequency were implemented, resulting in dense and sparse matrix representations, respectively.

The Space Formation sub-component implements cleanup options for the sparse word space, filtering low frequency words and links. Further development suggests pruning words, connector sets, and word-context links based on mutual information or other interaction information criteria.

Singular Value Decomposition (SVD) [13] can be applied to the sparse vector space to produce dimensionally-reduced dense vector representations (word embeddings). However, this approach provided unstable results when applying K-means clustering, so totally different distributions of words across clusters were formed with different random seeds. Using other clustering algorithms, no clusters were obtained at all.

An alternative approach was to project the filtered word space onto a vector space similar to multivariate Bernoulli distribution [14], with each word represented as a

sparse *j*-dimensional vector of binary variables. In this space, each variable describes the interaction between a word and a context (connector or connector-set), taking the values 1 if a word appears in a given context, or 0 otherwise. Exploring the properties of the resulting word space and whether these variables are correlated or dependent is an objective of further exploration. This approach smooths the influence of word frequencies and the distribution of interaction metrics on word vector similarities. However, preliminary studies have shown that rarely occurring words may have negative impact on the quality of space and consistency of the following results obtained from it. That means, more research is required to suppress such "noise" based on frequency filters.

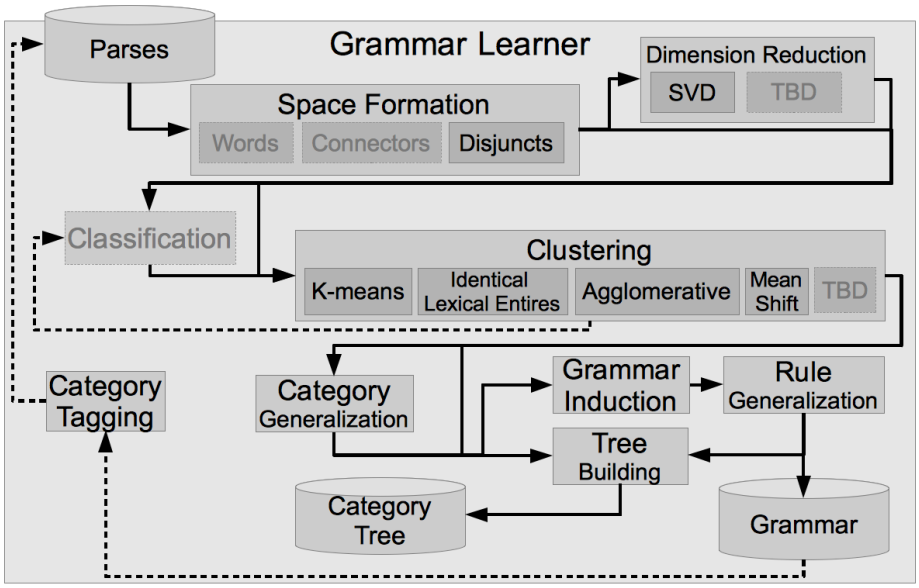

Fig. 2. Detailed architecture of the Grammar Learner component of the ULL pipeline. Grayed (dimmed) components of the architecture are designed but not currently implemented, and "TBD" blocks specify that new algorithms for a given stage of the process may be added in the future. Dashed lines indicate reverse flow direction, introducing loops in the pipeline.

The Clustering component may use various algorithms: beyond common K-means, the present research effort implemented and studied mean shift and agglomerative (ALE) clusterings, as well as grouping Identical Lexical Entries (ILE).

K-means clustering [15] of word embeddings used in our previous studies [4] turned out to introduce instability during the optimization of the entire pipeline parameters, so it was used only during earlier phases of the research. Also, our first results for mean shift clustering [16] were not significantly better than ALE. Hence, results for K-means and mean-shift clustering are not presented in the next section.

Agglomerative clustering in sparse vector space (implementation from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClus-

tering.html), further referred to as “ALE” (Agglomerating Lexical Entries), proved to be the best fit for larger datasets.

While testing similarity metrics for ALE, Euclidean distance provided better results for larger datasets than cosine and Manhattan distances. The clustering quality was evaluated with the Silhouette index for cosine, Jaccard, Euclidean, and Chebyshev similarities; cosine distance was preferred for smaller datasets. For larger corpora, all tested variations of the Silhouette index were close to zero, so no programmatic determination of the optimal number of clusters to create was possible (as opposed to our earlier work with K-means [4]). Therefore, we explored the target-clusters parameter space using 20, 50, 500, 1000, and 2000 clusters.

The ILE algorithm introduced in our previous work [2,4] actually implements lossless compression of a vector space by grouping words with the same sets of associated connectors or connector-sets into grammatical categories. The resulting space can be considered a straight projection of a fine-grained LG dictionary with the maximum number of word categories onto the space of connectors or connector-sets. However, ILE clustering creates very sparse LG dictionaries that could not be processed by the LG parser in its current version, due to combinatorial explosions and stack overflow issues in run-time.

Further development suggests iterative clustering process, involving incrementally increasing volume of input data from smaller amount of high-frequency words to larger amounts of less frequency words. In such case, the dimmed Classification component in Fig.2 could be used to attempt to classify newly experienced words to some of the categories learned from the previous iterations. Then, if some of words are not classified, they can be used to learn new clusters to be added to set of the categories. Still, exploration of the described flow has been not included in this study.

Category Generalization can be applied after Clustering for further aggregation of the learned word categories, based on Jaccard similarities of sets of connectors or connector-sets associated with them. Similarity thresholds can be set as generalization parameters; by gradually decreasing the threshold from the maximum found in the category distribution to a desired value, an iterative generalization process can be set up to provide hierarchical category trees showing the inner structure of categories agglomeration. Category Generalization results are not presented below, as Grammar Rules Generalization with the same algorithm demonstrated more efficiency.

The Grammar Induction component infers a grammar in the LG formalism [3] by processing links from input parses and replacing words with their corresponding learned word categories. Sets of links corresponding to each word, expressed in terms of word categories, form Link Grammar disjuncts for the category of the word. Link Grammar rules are sets of disjuncts associated with word categories.

Finally, the Grammar Rule Generalization component may be used to further cluster the learned word categories based on Jaccard similarities of sets of Link Grammar disjuncts associated with the categories in the Link Grammar rules. This component also adds an “upper layer” to the grammatical category tree on top of the “middle” layer representing word categories, which is anticipated to correspond to higher-level grammatical and semantic categories.

Optionally, the grammar learning process may be run in an iterative loop, using word categories from grammar rules found in a previous iteration as input to categorize words in subsequent iterations. The Category Tagging component replaces the words in input parses with learned categories (when available) so that more and more dense vector spaces may be created on subsequent iterations. The same iterative approach may be employed for incremental grammar learning, where the scope of the input parses gradually increases by adding previously unseen data to the training corpus.

3 Grammar Testing and Evaluation Metrics

The Grammar Tester (GT) component of the ULL pipeline implements a quality assurance procedure on the induced grammar obtained by the Grammar Learner. Two metrics are employed for this purpose: parse-ability and F1-score, as shown in Fig. 4.

The first quality criterion determines the extent to which the reference corpus is parsed at all – it is called "parse-ability" (PA) and computes the average percentage of words in a sentence recognized by the GT: $PA = (\Sigma(k_i/n_i))/N$, where N is the number of evaluated sentences, k_i is the number of words in the *i-th* sentence recognized by the GT, and n_i is the total number of words in *i-th* sentence.

As a second metric, we use the conventionally defined F-measure or F-score (*F1*), a function of recall (R) and precision (P): F1 = 2*R*P/(R+P). Recall is defined as $R=(\Sigma(c_i/e_i))/N$, and precision as $P=(\Sigma(c_i/l_i))/N$, where c_i is the number of correctly identified links in *i-th* sentence, e_i is the number of expected links and l_i is the number of identified links, including false positives. That is, for recall we take the average per-sentence number of overlapping links in test and reference parses divided by the total number of links in the reference parses. Respectively, for precision we take the same overlapping number, divided by the total number of links in the test parses.

4 Methodology of Studies and Intermediate Results

Our experiments for the ULL pipeline were performed with the three English text corpora referenced in earlier work [4] and presented on Fig.3: 1) an artificial corpus created for basic testing purposes, the Proof-of-Concept English (POCE) corpus; 2) the Child Directed Speech (CDS) corpus obtained from subsets of the CHILDES corpus – a collection of English communications directed to children with limited lexicon and grammar complexity (https://childes.talkbank.org/derived/) [17,18,19]; 3) the Gutenberg Children (GC) corpus – a compendium of books for children contained within Project Gutenberg (https://www.gutenberg.org), following the selection used for the Children's Book Test of the Babi CBT corpus [14] (https://research.fb.com/downloads/babi/).

For each of these corpora, we ran our Grammar Learner using two different kinds of parses as input: first, our "standard" parses created either manually (for the POCE corpus), or parsed by the LG parser using the standard human-crafted Link Grammar Dictionary for the English language – further called LG-Parses. The second type of

parses used are MST-Parses created by the previous segment of the ULL pipeline, including parses with WSD applied [4]. The human-knowledge-based LG-parses were used as a reference to asses the quality of MST-parses, as well as to create a baseline input for the GL to gauge its ability to induce grammar from "ideal" parses.

Corpus	Total words	Unique words	Occurrences per word	Total sentences	Average sentence length
POC-English	388	55	7	88	4
Child-Directed Speech	124185	3399	37	38181	4
Gutenberg Children	2695151	54054	50	207130	13

Fig. 3. Some features of the English text corpora used for studies. See [4] for more details.

Corpus	Parses	Parses F1	Clustering	Grammar PA	Grammar F1
POC-English	Manual	1.0	ILE	100%	1.0
POC-English	Manual	1.0	ALE-400	100%	1.0
POC-English	MST	0.71	ILE/G	100%	0.72
POC-English	MST	0.71	ALE-400	100%	0.73
Child-Directed Speech	LG	1.0	ILE	99%	0.98
Child-Directed Speech	LG	1.0	ALE-400	99%	0.97
Child-Directed Speech	MST	0.68	ILE/G	71%	0.45
Child-Directed Speech	MST	0.68	ALE-400/G	82%	0.50
Gutenberg Children	LG	1.0	ALE-50	90%	0.61
Gutenberg Children	LG	1.0	ALE-500	56%	0.55
Gutenberg Children	MST	0.52	ALE-50	N/A	N/A
Gutenberg Children	MST	0.52	ALE-500	81%	0.48

Fig. 4. Best scores for F-measure (F1) and parse-ability (PA) for different corpora and parse types using different clustering algorithms: ILE – Identical Lexical Entries, ILE/G – ILE with Grammar Rule Generalization, ALE-400 – Agglomerative clustering for 400 target categories, ALE-400/G – same with Grammar Rule Generalization, ALE-50 and ALE-500 – Agglomerative clustering for 50 and 500 target categories, respectively.

Based on the study of the various configurations of the Grammar Learner with different parses for each given corpora, and having generated approximately 100 induced grammars and evaluated them as specified before, we present the best results obtained in Fig 4. From this experience, the following observations can be made:

1. It is possible to perform grammar learning using a non-dimensionally-reduced discrete vector space of lexical entries (Link Grammar "disjuncts") without dimension

reduction, based on identical lexical entries (ILE clustering) and using agglomerative clustering for English corpora of different scales, achieving reasonable scores for parse-ability and F-measure (PA/F1), using either parses obtained with English Link-Grammar dictionary (LG-parses) or MST-parses, as shown on Fig.4.

2. It has been found that for real-world corpora such as CDS and GC, better PA/F1 scores are obtained if the evaluation of the grammar is performed only for sentences for which the LG-parses are complete (the LG parser is allowed to ignore some words from a parse if a "cheaper" parse tree is found with an incomplete sentence). This way, the comparison is less likely to be done against originally incorrect parses.

3. We found a large Pearson correlation (93-100%) between the distributions of F-scores of MST-parses against LG-parses, and that of the parses obtained based on the grammar induced from these MST-parses against the same LG-parses. This effectively means that the quality of a learned grammar is linearly related to the quality of its input parses.

4. No reliable correlation between PA and F1 was found across corpora: in some cases (POCE, cleaned version of CDS and raw GC) it is positive, for another one it is close to zero (raw CDS), and for a third one it is negative (cleaned version of GC). That means PA can not be used as a metric for hyper-parameters optimization when we lack a standard (like LG-parses) to measure F1.

5. It has been shown that applying word-sense disambiguation before MST-parsing can improve the parses, providing higher F1 against their LG-parses standard. For the POCE corpus, F1 (on MST-parses only, not from grammar induction) improves from 0.70 to 0.75; in the case of the GC corpus, it grows from 0.50 to 0.52. As predicted by the point 3 above, the quality of the learned grammar increases as well.

6. The results shown in Fig. 5 were achieved by either grouping identical lexical entries (ILE) or agglomerative clustering (ALE), both starting from a discrete vector space of lexical entries (Link Grammar disjuncts) without dimension reduction. These results replace the ones from previously-used dimension reduction with singular value decomposition (SVD) and K-means clustering. Such changes provide higher F-scores and reproducibility, allowing optimization of the pipeline's hyper-parameters.

7. It has been found that the number of clusters, representing grammatical categories, that provides the best F1 for produced parses is about 500 for the real-world English corpora (CDS and GC). A decrease in the number of clusters/categories tends to increase PA and decrease F1 rapidly; using more clusters tends to reduce both PA and F1 slowly. Also, inducing grammars with less than 50 categories on the GC corpus causes exponential run-time growth for the LG parser using them, as well as segmentation faults on particular sentences.

8. We noticed that removing parses with low-frequency words from the GL input may decrease the grammar induction run-time, but never increase quality (either PA or F1) given our corpora; literally "the more words, the better".

9. Figure 5 shows that it is possible to use generalization of the learned grammatical categories into hierarchical trees to unravel the grammatical and semantic nature of their vocabulary in a reasonable way, corresponding to the context of the training corpora. These categorical trees can be useful for feature engineering in NLP applications, as well as for studies of new languages or domains by computational linguists.

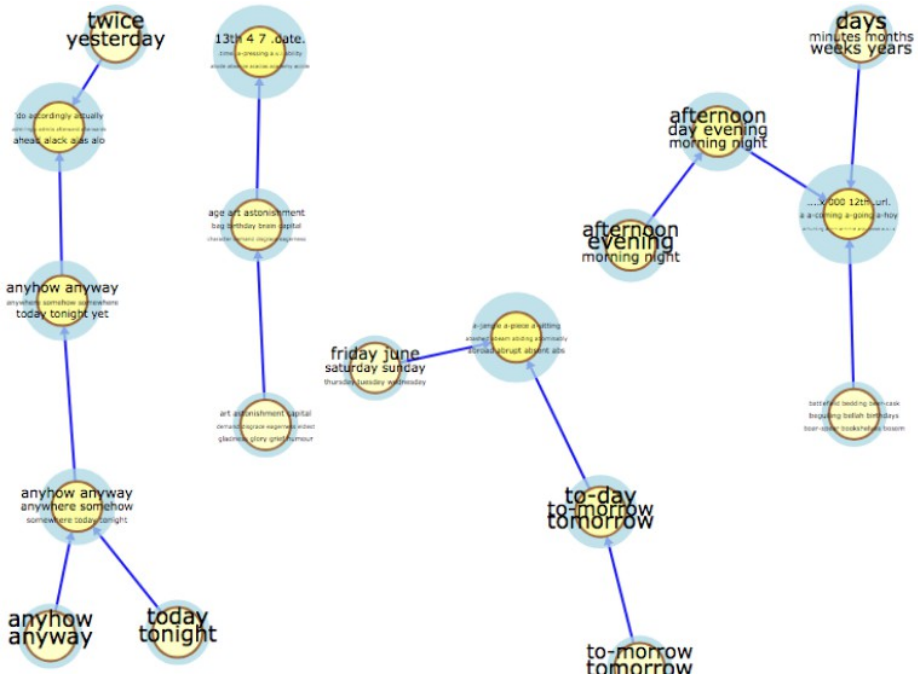

Fig. 5. Fragment of a category tree learned from the Gutenberg Children corpus in an unsupervised way, showing subgraphs matching the word "day". Visualized with the Aigents Graphs framework (https://github.com/aigents/aigents-java/blob/master/html/graphs.html).

5 Conclusion

We can conclude that it is generally possible to perform programmatic unsupervised induction of formal grammars from unannotated sentence parses for tiny, small and large text corpora, using the Link Grammar formalism and parser. We have found that quality of the grammar is linearly correlated with quality of the input parses used to induce the grammar. That is, the quality of the input parses seems to be the major obstacle for obtaining high quality grammars.

Future plans for our work include searching for ways to improve the quality of the input parses obtained in an unsupervised way from unannotated text corpora, as well as enhancing the grammar-induction technology itself. For the latter, we intend to improve the GL component to learn generalized parts of speech and grammatical relationships through better clustering.

6 Acknowledgements

We appreciate contributions by Linas Vepstas, including insightful discussions and critique on our research. We thank Amir Plivatsky for valuable feedback and maintenance and incremental improvements of the LG parser technology used in our work.

References

1. Yuret D.: Discovery of Linguistic Relations Using Lexical Attraction. arXiv:cmp-lg/9805009 [cs.CL] (1998).
2. Vepstas L., Goertzel B.: Learning Language from a Large (Unannotated) Corpus. arXiv:1401.3372 [cs.CL] 14 Jan 2014 (2014).
3. Sleator D., Temperley D.: Parsing English with a Link Grammar. Third International Workshop on Parsing Technologies (1993).
4. Glushchenko A., Suarez A., Kolonin A., Goertzel B., Castillo C., Leung M. H., Baskov O.: Springer Lecture Notes in Computer Science book series (LNCS, volume 10999). Unsupervised Language Learning in OpenCog. AGI 2018: Artificial General Intelligence, pp 109-118 (2018).
5. Wrenn J., Stetson P., Johnson S.: An Unsupervised Machine Learning Approach to Segmentation of Clinician-Entered Free Text. AMIA Annu Symp Proc. 2007; 2007: 811–815. (2007).
6. Castillo-Domenech C., Suarez-Madrigal A.: Statistical parsing and unambiguous word representation in OpenCog's Unsupervised Language Learning project. Göteborg : Chalmers University of Technology (2018). https://publications.lib.chalmers.se/records/fulltext/256408/256408.pdf
7. Dupoux E: Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. arXiv:1607.08723 [cs.CL] (2018).
8. Goertzel B., Pennachin C., Geisweiller N: Engineering General Intelligence, Part 2: The CogPrime Architecture for Integrative. Embodied AGI, ATLANTIS PRESS (2014).
9. Harwath D., Torralba A., Glass J.: Unsupervised Learning of Spoken Language with Visual Context. 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain. (2016).
10. Došilović F., Brčić M., Hlupić N.: Explainable artificial intelligence: A survey. Published in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (2018).
11. Levy, O., Goldberg. Y.: Neural Word Embedding as Implicit Matrix Factorization. NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, vol.2, pp 2177-2185 (2014)
12. Church K., Hank P.: Word association norms, mutual information, and lexicography. Computational Linguistics archive, vol.16, issue 1, pp 22-29 (1990)
13. Wall M., Rechtsteiner A., Rocha L.: Singular Value Decomposition and Principal Component Analysis, arXiv:physics/0208101 (2002)
14. Dai, B., Ding, S., Wahba, G.: Multivariate Bernoulli distribution. Bernoulli, vol.19, no.4, pp 1465-1483 (2013)
15. Sculley D., Web-scale k-means clustering, WWW '10 Proceedings of the 19th international conference on World-wide-web, pp. 1177-1178, Raleigh, NC, USA (2010)
16. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, issue 5, pp 603-619 (2002)
17. Bernstein-Ratner N.: The phonology of parent child speech. Children's language, 6(3). (1987).
18. Brent M., Cartwright T: Distributional regularity and phonotactic constraints are useful for segmentation. Cognition, 61:93–125. (1996).
19. Brent M., Siskind J.: The role of exposure to isolated words in early vocabulary development. Cognition, 81(2):B33–B44. (2001).