Doctoral Dissertation

EVOLUTIONARY MODEL(S) OF ONTOGENY OF LINGUISTIC CATEGORIES

FOUR SIMULATIONS

BY MSC. BC. ET BC. DANIEL HROMADA

In order to obtain a degree *Philosophiae Doctor*

from

Institute of Robotics and Cybernetics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology, Bratislava (STU)
Study program: 9.2.7 Cybernetics

and

EA 4004 - CHArt - Cognitions Humaine et Artificielle
Doctoral School 'Cognition, Langage, Interaction'
University Paris Lumières (P8)
Specialization: Cognitive Psychology

Year of deposit: 2016
Date and time of defense : 5.9.2016 at 10:00

Place of defense:
Institute of Robotics and Cybernetics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology
Bratislava, Slovak Republic, European Union

prof. Charles Tijus	Thesis director	P8
Ivan Sekaj doc. Ing. PhD.	Thesis director	STU

Daniel Hromada: *Evolutionary Model(s) of Ontogeny of Linguistic Categories,* Propedeutica Didactica II., © June 2016

PUBLICATIONS

Some ideas have appeared previously in the following publications:

El Ghali, A., Hromada, D., and El Ghali, K. (2012). Enrichir et raisonner sur des espaces sémantiques pour l'attribution de mots-clés. In *Actes de l'atelier de clôture du huitième défi fouille de texte (DEFT)*, pages 81–94, Grenoble, France.

Hromada, D. D. (2010a). Quantitative intercultural comparison by means of parallel pageranking of diverse national wikipedias. In *10th International Conference on the Statistical Analysis of Textual Data-JADT 2010*, pages 643–651, Rome, Italy. Edizioni Universitarie di Lettere Economia Diritto.

Hromada, D. D. (2010b). smiled : Sourire naturel et sourire artificiel. de l'utilisation d'opencv pour le tracking, la reconnaissaince des expressions faciales et la détection du sourire. Master's thesis, Ecole Pratique des Hautes Etudes, Paris, France.

Hromada, D. D. (2011a). *The Central Problem of Roboethics: from Definition towards Solution*. MV-Wissenschaft.

Hromada, D. D. (2011b). Initial experiments with multilingual extraction of rhetoric figures by means of perl-compatible regular expressions. In *RANLP Student Research Workshop*, pages 85–90, Borovec, Bulgaria.

Hromada, D. D. (2013a). Parallel democracy model and its first implementations in the cyberspace. *Teoria politica*, 3:165–180.

Hromada, D. D. (2013b). Random projection and geometrization of string distance metrics. In *RANLP Student Research Workshop*, pages 79–85, Borovec, Bulgaria.

Hromada, D. D. (2013c). Thesis for rigorous examination. Defended on 4.11.2013 at Slovak University of Technology in Bratislava.

Hromada, D. D. (2014a). Comparative study concerning the role of surface morphological features in the induction of part-of-speech categories. In *Text, Speech and Dialogue*, pages 46–52, Brno, Czech Republic. Springer.

Hromada, D. D. (2014b). Conditions for cognitive plausibility of computational models of category induction. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 93–105, Montpellier, France. Springer.

Hromada, D. D. (2014c). Empiric introduction to light stochastic binarization. In *Text, Speech and Dialogue*, pages 37–45, Brno, Czech Republic. Springer.

Hromada, D. D. (2015a). *Conceptual Foundations : Intramental Evolution & Ontogeny of Toddlerese*. Propedeutica Didactica. in print. Supplementary material for PhD. dissertation.

Hromada, D. D. (2015b). Genetic optimization of semantic prototypes for multiclass document categorization. In *Proceedings of Elitech 2015 conference*, Bratislava, Slovak Republic. Slovak University of Technology. Awarded "best paper" prize in "Applied Informatics" track.

Hromada, D. D. (2016a). Narrative fostering of morality in artificial agents: Constructivism, machine learning and story-telling. In *L'esprit au-delà du droit: Pour un dialogue entre les sciences cognitives et le droit*. Mare et Martin.

Hromada, D. D. (2016b). Reproducible identification of pragmatic universalia in childes transcripts. In *Proceedings of 13th International Conference on Statistical Analysis of Textual Data*, pages 541–550. Universite Nice Sophia-Antipolis, France.

Hromada, D. D. and Gaudiello, I. (2014). Introduction to moral induction model and its deployment in artificial agents. In *Sociable Robots and the Future of Social Relations*, pages 209–216. IOS Press.

Hromada, D. D., Tijus, C., Poitrenaud, S., and Nadel, J. (2010). Zygomatic smile detection: The semi-supervised haar training of a fast and frugal system: A gift to opencv community. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pages 241–245, Hanoi, Vietnam. IEEE.

Many fulltexts (+ some other materials) are available for download at http://wizzion.com/papers

BibTeXpublication list is available at http://wizzion.com/papers/hromada-publications.bib

HOW IS THIS DISSERTATION ORGANIZED

In its essence, this dissertation is a collection of four scientific articles. Each article describes a distinct simulation and can be read individually. Each article is preceded by a "generic introduction" and followed by a "generic conclusion". These aim to ground the article into overall context of the dissertation.

Taken together, an article and its general introduction form a chapter. Chapters devoted to four simulations are followed by *Summa* which aspires to subsume all simulations under a common *clef-de-voute* furnished by the theory of Intramental Evolution. Each chapter is followed by its proper bibliography.

REFERENCES TO CONCEPTUAL FOUNDATIONS

Dissertation often implements expert terminology issued from disciplines as diverse as computer science, linguistics, psychology, natural language processing, biology, neurosciences, thermodynamics, philology, anthropology et caetera. In order to aleviate potential misunderstandings, some terms are immediately followed by an expression **P+X** whereby X is a variable of type integer.

Whenever such an expression is encountered, P+ is to be interpreted as a marker indicating a reference to Conceptual Foundations (i.e. Hromada (2015)) and the value of X as a page number of Conceptual Foundations (CF) where the preceding term is precisely defined or at least more closely discussed. For example, reader is invited to interpret the token "learning (P+4)" as "c.f. (Hromada, 2015, pp.4) to see the definition of the term *learning*".

References to discussions spanning multiple pages are also possible. In this case, marker shall have a form **P+X-Y** whereby X denotes the page number where discussion begins and Y denotes the page number where discussion ends. For example, reader is invited to interpret the token "hard thesis (P+3-10)" as meaning "please read (Hromada, 2015, pp.3-10) to share author's notion of the expression *hard thesis*".

Because of these reasons, it is recommended to have the volume of CF at hand during a more profound lecture of this dissertation.

ASPIRATIONS AND COMMONALITIES AMONG 4 SIMULATIONS

All simulations have one thing in common: their ultimate aspiration is to provide different facets of "cognitively plausible (P+13)", *ex compu-*

tatio et simulatio proof-of-concept for a theory of intramental evolution (P+3) introduced in Hromada (2015).

There are other characteristics shared by all simulations:

- inputs are linear sequences of discrete graphemic symbols (i.e. "text", P+20)
- they aim to offer solution(s) to problem(s) of essentially linguistic nature
- they implement "Evolutionary Computation" (P+11) methods to solve such problems

One can observe further common characteristics among first, second and third simulation. They all implement

- projection of all textual entities onto euclidean vector spaces (P+131-136) by means of random-indexing (P+138-139)
- transformation of such 128-dimensional euclidean spaces into 128-dimensional binary (Hamming) spaces
- formal definition that a category is a Hamming ball (i.e. an N-dimensional convex set defined by its centroid and radius)
- evolutionary search for ideal constellations of such categories within their respective binary spaces

We shall sometimes use the term "category-inducing" (CI) simulation to refer to these three simulations.

All simulations - i.e three CI-simulations as well as the zeroth simulation - are implemented in the programming language PERL[1]

AMBITIONS OF INDIVIDUAL SIMULATIONS

Each individual simulation has its individual ambition. Hence:

- zeroth simulation aspires to demonstrate that Evolutionary Computation (EC) can offer useful insights to an agent hoping to break the code of an unintelligible corpus (e.g. to help decode riddle as cryptic as the Voynich Manuscript)
- first simulation aspires to demonstrate that EC can be a useful means of multiclass classification of textual documents according to their semantic content (e.g. and in a Big Data scenario could potentially lead to results as good as those produced by connectionist "deep learning" methods)

[1] PERL code is often less ambigous and hence more reproducible (Hromada, 2016b) than a so-called "pseudocode". For this reason(s), source code snippets included into this dissertation are presented in PERL and not in the pseudocode.

- second simulation aspires to demonstrate that EC can help to identify useful solutions to problem of multiclass part-of-speech classification
- third simulation aspires to demonstrate that EC can pave the way to induction (P+148-162) of plausible micro-grammars from solely positive corpus of motherese (P+91) utterances

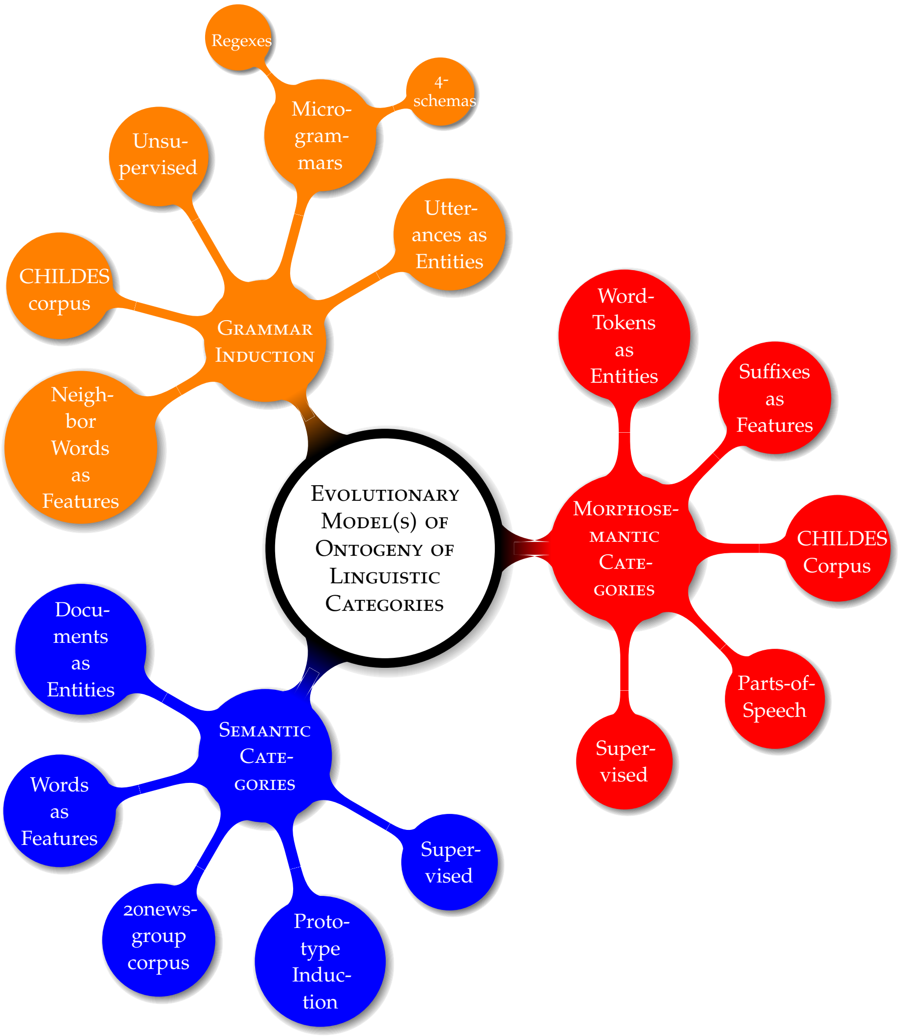

Figure 1: Distinctions among category-inducing (CI) simulations.

When compared with mutually-aligned CI-simulations, the zeroth simulation could be potentially regarded as somewhat "exotic". It could potentially even be remarked that since it does not involve any induction of categories whatsoever, it should not be included in the dissertation aiming to elucidate evolutionary principles behind the ontogeny of linguistic categories.

In spite of this, we have decided - after having obtained the encouragement of both Thesis directors - to include the zeroth simulation in this work. For it was indeed the challenge posed by the zeroth simulation which made us program our first evolutionary algorithm.

0 BREAKING INTO UNKNOWN CODE

0.1 Generic introduction

A cryptologue posed with an unbroken cipher is, in certain sense, in a position similar to a child (P+19) which has just been born into our common world. Both cryptologue and a child are confronted with novel constellations of symbols and features. Both assume that the data with which they are confronted - a motherese (P+90-93) utterance perceived by a child or a cipher studied by a cryptologue - ultimately carry a certain meaningful message. Both combine their ingenuity with relentless perseverance: both accept that the path to success leads through ocean of trials and errors (P+22). Ultimately, they both transcend their initial state of limited knowledge and attain understanding: child shall understand the world and the scholar shall understand the cipher.

This analogy between a child and a cipher-breaker can be pushed even further in case we speak about the cipher stored in the enigmatic medieval Voynich Manuscript (VM). This is so because VM contains a non-negligible amount of *visual content* and it can be rightfully speculated that if VM contains a cipher to be decoded, than the deciphering process (and its subsequent evaluation) shall be founded on discovery of associations between VM's visual content and the adjacent "voynichese" script.

This is - we believe - similar to the position of a visually non-impaired human child who acquires a non-negligible amount of information about her world and her language by means of associating the components of surrounding visual scenes with simultaneously heard phonemic sequences (e.g. "red ball in mama's hand").

This being said, let's now present first implications of our "child as a cryptologue" analogy, as published in the article Hromada (2016).

0.2 Abstract

Voynich Manuscript is a corpus of unknown origin written down in unique graphemic system and potentially representing phonic values of unknown or potentially even extinct language. Departing from the postulate that the manuscript is not a hoax but rather encodes authentic contents, our article presents an evolutionary algorithm which aims to find the most optimal mapping between voynichian glyphs and candidate phonemic values.

Core component of the decoding algorithm is a process of maximization of a fitness function which aims to find most optimal set of substitution rules allowing to transcribe the part of the manuscript - which we call the Calendar - into lists of feminine names. This leads to micrgrammars which allow us to consistently transcribe dozens among three hundred calendar tokens into feminine names: a result far surpassing both "popular" as well as "state of the art" tentatives to crack the manuscript. What's more, by using name lists stemming from different languages as potential cribs, our "adaptive" method can also be useful in identification of the language in which the manuscript is written.

As far as we can currently tell, results of our experiments indicate that the Calendar part of the manuscript contains names from baltoslavic, balkanic or Hebrew language strata. Two further indications are also given: primo, highest fitness values were obtained when the crib list contains names with specific in-fixes at token's penultimate position as is the case, for example, for Slavic feminine diminutives (i.e. names ending with -ka and not -a). In the most successful scenario, 240 characters contained in 35 distinct Voynichese tokens were successfully transcribed. Secundo, in case of crib stemming from Hebrew language, whole adaptation process converges to significantly better fitness values when transcribing voynichian tokens whose order of individual characters have been reversed, and when lists feminine and not masculine names are used as the crib.

0.3 introduction

Voynich Manuscript (VM) undoubtedly counts among the most famous unresolved enigmas of the medieval period. On approximately 240 vellum pages currently stored as manuscript (MS) 408 in Yale University's Beinecke Rare Book and Manuscript Library, VM contains many images apparently related to botanics, astronomy (or astrology) and bathing. Written aside, above and below these images are bulks of sequences of glyphs. All this is certain.

Also certain seems to be the fact that in 1912, VM was re-discovered by a polish book-dealer Wilfrid Voynich in a large palace near Rome called Villa Mandragone. Alongside the VM itself, Voynich also found the correspondence - dating from 1666 - between Collegio Romano scholar Athanasius Kircher and the contemporary rector of Charles University in Prague, Johannes Marcus Marci. Other attested documents - e.g. a letter from 1639 sent to Kircher by a Prague alchemist Georg Baresch - also indicate that during the first half of 17th century, VM was to be found in Prague. The very same correspondence also indicates that VM was acquired by famous patron of arts, sciences and alchemy, Emperor Rudolf II. [1]

Asides this, one more fact can be stated with certainty: the vellum of VM was carbon-dated to the early 15h century (Hodgins, 2014).

0.3.1 PRE-DIGITAL TENTATIVES

Already during the pre-informatic era of first half of 20th century had dozens, if not hundreds, men of distinction invested non-negligible time of their life into tentatives to decipher the "voynichese" script.

Being highly popular in their time, many such tentatives - like that of Newbold who claimed to "prove" that VM was encoded by Roger Bacon by means of 6-step anagrammatic cipher (Newbold, 1928b), or that of Strong (Strong, 1945) who claimed VM to be a 16th-century equivalent of the Kinsey Report" - may seem to be, when looked upon through the prism of computer science, somewhat irrational [2].

C.f. (d'Imperio, 1978) for a overview of other 20th-century "manual" tentatives which resulted in VM-deciphering claims. After description of these tentatives and and after presentation of informationally very rich introduction to both VM and its historical context, d'Imperio adopts a skeptical stance towards all scholars who associated VM's origin with the personage of Roger Bacon[3].

In spite of skeptic who she was, d'Imperio hadn't a priori disqualified a set of hypotheses that the language in which the VM was ultimately written was Latin or medieval English. And such, indeed, was the majority of hypotheses which gained prominence all along 20th century.[4].

0.3.2 POST-DIGITAL TENTATIVES

First tentatives to use machines to crack the VM date back to prehistory of informatic era. Thus, already during 2nd world war did the cryptologist William F. Friedman invited his colleagues to form

[1] Savants which passed through Rudolf's court included Johannes Kepler, Tycho de-Brahe or Giordanno Bruno. The last one is known to have sold a certain book to the emperor for 600 ducats.

[2] Note, for example, Strong's "translation" of one VM passage: "*When the contents of the veins rip, the child comes slyly from the mother issuing with leg-stance skewed and bent while the arms, bend at the elbow, are knotted like the legs of a craw-fish.*" Strong (1945) Note also that such translation was a product of man who was "a highly respected medical scientist in the field of cancer research at Yale University" (d'Imperio, 1978).

[3] "I feel, in sum, that Bacon was not a man who would have produced a work such as the Voynich manuscript...I can far more easily imagine a small society perhaps in Germany or Eastern Europe (d'Imperio, 1978, 51)"

[4] Note that such pro-English and pro-Latin bias can be easily explained not by the properties of VM itself, but by the simple fact that first batches of VM's copies were primarily distributed and popularized among Anglosaxon scholars of medieval philosophy, classical philology or occidental history

"extracurricular" VM study group - programming IBM computers for sorting and tabelation of VM data was one among the tasks. Two decades later - and already in position of a first chief cryptologist of the nascent National Security Agency - Friedman had formed the 2nd study group. Again without ultimate success.

One member of Friedman's 2nd Study Group After was Prescott Currier whose computer-driven analysis led him to conclusion that VM in fact encodes two "statistically distinct" (Currier, 1970) languages. What's more, Currier seems to have been the first scholar who facilitated the exchange and processing of Voynich manuscript by proposing a transliteration[5] of voynichese glyphs into standard ASCII characters. This had been the predecessor of the European Voynich Alphabet (EVA) (Landini and Zandbergen, 1998) which had become a de facto standard when it comes to mapping of VM glyphs upon the set of discrete symbols.

Canonization of EVA combined with dissemination of VM's copies through Internet have allowed more and more researchers to transcribe the sequence of glyhps on the manuscript into ASCII EVA sequences. Is is thanks to laborious transcription work of people like Rene Zandberger, Jorge Stolfi or Takeshi Takahashi that verification or falsification of VM-related hypotheses can be nowadays in great extent automatized.

For example, Stolfi's analyses of frequencies of occurrence of different characters in different contexts has indicated that majority of Voynichese words seems to implement a sort of tripartite crust-core-mantle (or prefix, infix, suffix) morphology. Later study has indicated that the presence of such morphological regularities could be explained as an output of a mechanical device called Cadran grill (Rugg, 2004). The "hoax hypothesis" is also supported by the study of Schinner (2007) who suggested that "the text has been generated by a stochastic process rather than by encoding or encryption of language". Pointing in the similar direction, the analysis also concludes that "glyph groups in the VM are not used as words".

On the other hand, a methodology based on "first-order statistics of word properties in a text, from the topology of complex networks representing texts, and from intermittency concepts where text is treated as a time series" presented in (Amancio et al., 2013) lead its authors to conclusion that VM "is mostly compatible with natural languages and incompatible with random texts". Simply stated, the way how diverse "words" are distributed among different sections of VM indicates that these words carry certain semantics. And this indicates that VM, or at least certain parts of it, are not a hoax.

[5] In this article we distinguish transliteration and transcription. Transliteration is a bijective mapping from one graphemic system into another (e.g. VM glyphs is transliterated into ASCII's EVA subset). Transcription is a potentially non-bijective mapping between symbols one one side and sound- or meaning- carrying units on the other.

0.3.3 our position

Results of (Amancio et al., 2013) had made us adopt the conjecture "VM is not a hoax" as a sort of a fundamental hypothesis accepted *a priori*. Surely, as far as we stand, it could not be excluded that VM is a work of an abnormal person, of somebody who suffered severe schizophrenia or was chronically obsessed by internal glossolalia (Kennedy and Churchill, 2005). Nor can it be excluded that the manuscript does not encode full-fledged utterances but rather lists of indices, sequences or proper names of spirits-which-are-to-be-summoned or sutra-like formulas compressed in a sort of private pidgin or a sociolect. But given VM's ingenuity and given the effort which the author had to invest into the conception of the manuscript and given a sort of "elegant simplicity" which seems to permeate the manuscript, we have felt, since our very first contact with the manuscript, a sort of obligation to interpret its contents as meaningful.

That is, as having the capability of denoting the objects outside of the manuscript itself. As being endowed with the faculty of reference to the world (Frege, 1994) which we, 21st century interpreters, still inhabit hundred years after VM's most plausible date of conception.

It is with such bias in mind that our attention was focused upon a certain regularity which we have later decided to call "the primary mapping".

0.3.4 primary mapping

Condition sine qua non of any act of deciphering is a discovery of rules which allow to transform initially meaningless cipher into meaningful information. In most trivial case, such deciphering is facilitated by a sort of Rosetta Stone (Champollion, 1822) which the decipherer already has at his disposition. Since both the cipher-text as well as the plain-text (also called "the crib") are explicitly given by the Rosetta Stone, discovery of the mapping between the two is usually quite straightforward.

The problem with VM is, of course, that it seems not to contain any explicit key which could help us to decipher its glyphs. Thus, the only source of information which could potentially help us to establish reference between VM's glyphs and the external world are VM's drawings. One such drawing present atop of folio f84r is shown on Figure 2.

Figure 2 displays twelve women bathing in eight compartments of a pool. Bathing women is a very common motive present in VM and there seems to be nothing peculiar about it. The fact that word-like sequences are written above heads of these women is also trivial.

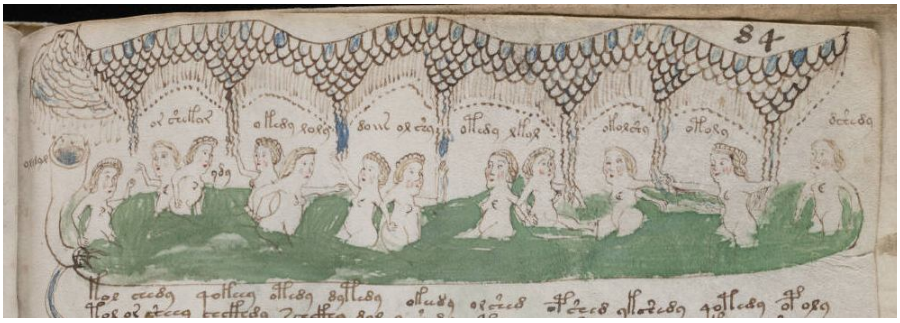

Figure 2: Drawing from folio f84r containing the primary mapping.

One can, however, observe one regularity which seems to be interesting. That is, in case two women bath in the same compartment, the compartment contains two word-like sequences. If one woman bathes in the compartment, there is only one word-like sequence which is written above her head.

One figure - one word, two figures - two words. This principle is stringently followed and can be seen on other folios as well. What is more, the words themselves are sometimes similar but they are not the same. Such trivial observations lead to trivial conclusion: these word-like sequences are labels.

And since these names are juxtaposed to feminine figures, it seems reasonable to postulate that these labels are, in fact, feminine names. This is the primary mapping.

0.3.5 three conjectures

Method which shall be described in following sections can be considered as valid only under assumption that following conjectures are valid:

1. "the primary mapping conjecture" : voynichese words asides feminine figures are feminine names
2. "diachronic stability of proper names" : proper names are less prone to diachronic change than other language units
3. "Occam razor" : instead of containing a sophisticated esoteric cipher, VM simply transmits a text written in an unknown script

Further reasons why we consider "the primary mapping conjecture" as valid shall be given alongside our discussions of "the Calendar". When it comes to conjecture postulating the "diachronic stability of proper names", we could potentially refer to certain cognitive peculiarities or how human mind tends to treat proper names (Imai and Haryu, 2001). Or focus the attention of the reader to the fact that for practically every human speaker, one's own name undoubtedly belongs among the most frequent and most important tokens which one hears or utters during whole life. This results in a sort of stability against linguistic change and allow the name to cross the centuries with higher probability than words of lesser importance and frequency.

But instead of pursuing the debate in such a direction, let's just point out that successful decoding of Mycenaean Linear script B ((Ventris and Chadwick, 1953) would be much more difficult if certain toponyms like *Amnisos, Knossos or Pylos* haven't succeeded to carry their phonetic skeleton through aeons of time.

At last but not least, the "Occam razor conjecture" simply explicitates the belief that **a reasonable scientist should not opt to explain VM in terms of anagrams and opaque hermeneutic procedures if similar - or even more plausible - results can be attained when approaching VM as it was a simple substitution cipher.**

0.4 METHOD

The core of our method is an optimization algorithm which looks for such a candidate transcription alphabet A_x which, when applied upon the list of word types occurring in VM's Calendar section yields an output list whose members should be ideally present in another list, called the Crib. The optimization is done by an evolutionary strategy - an individual chromosome encode a candidate transcription alphabet and a fitness function is given as a sum of lengths of all tokens which were successfully transcribed from Calendar to a specified Crib.

0.4.1 CALENDAR

Six among twelve words present on Figure 1. occur only on folio f84r. Six others occur on other folios as well, and five of these six words occur also as labels near feminine figures displayed on 12 folios of the section commonly known as "Zodiac". It is like this that our attention was focused from the limited corpus of "primary mapping" towards more exhaustive corpus contained in the Zodiac.

Every page of Zodiac displays multiple concentric circles filled with feminine figures. Attributes of these figures differ - some hold torches, some do not, some are bathing, some are not - but one pattern is fairly regular. Asides every woman there is a star and asides every star, there is a word.

While some authors postulate that these words are names of stars or names of days, we postulate that these words are simply feminine names[6]. From Takahashi's transliterations of twelve folios of the Zodiac we extract 290 tokens which instantiate 264 distinct word types.

To evit possible terminological confusion, we shall denote this list of 264 labels[7] with the term Calendar. Hence, Zodiac is the term to refer to folios f70v2 - f73v, while Calendar is simply a list of 264 labels. Total length of this 264 labels is 2045 letters. These characters are chosen from 19-symbol ($|A_{cipher}| = 19$) subset of the EVA transliteration alphabet.

0.4.2 CRIBBING

Cribbing is a method by means of which a hypothesis, that the Calendar contains lists of feminine names, can potentially lead to deciphering of the manuscript. For if the Calendar is indeed such a list, then one could use lists of existing and attested feminine names as hypothetical target "cribs".

In crypt-analytic terms, an intuition that the Calendar contains feminine names makes it possible to perform a sort of known-plain-text attack (KPA). We say "a sort of", because in case of VM are the "cribs" upon which we shall aim to map the Calendar, not known with 100% certainty. Hence, it is maybe more reasonable to understand the cribbing procedure as the plausible-plain-text attack (PPA).

This beings said, we label as "cribbing" a symbol-substituting procedure $P_{cribbing}$ which replaces symbols contained in the cipher (i.e. in the Calendar) with symbols contained in the plain-text. Hence, not only cipher but also plain-text are inputs of the cribbing procedure.

Every act of execution of $P_{cribbing}$ can be followed an act of evaluation of usefulness $P_{cribbing}$ in regards to its inputs. The ideal procedure would result in a perfect match between the rewritten cipher and the plain-text, i.e.

$$P_{cribbing}(cipher) == plain - text$$

On the other hand, a completely failed $P_{cribbing}$ results in two corpora which do not have anything in common.

And between two extremes of the spectrum, between "the ideal" and "the completely failed", one can place multitudes other procedures, some closer to the ideal than the others.

This makes place for optimization.

[6] It cannot be excluded, however, that they all this at once. Note, for example, that in many central European countries, it is still a fairly common practice to attribute specific names to specific days in a year, i.e. "meniny".

[7] Available at http://wizzion.com/thesis/simulation0/calendar.uniq

Listing 1: Discrete cross-over

```
#discrete crossover
my $child_genome;
my $i=0;
for (@mother_genome) {
        if ($_ ne $father_genome[$i]) {
                rand > 0.5 ? ($child.=$mother_genome[$i]) : (
                    $child.=$father_genome[$i]);
        } else {
                $child_genome.=$mother_genome[$i];
        }
        $i++;
}
```

0.4.3 OPTIMIZATION

All experiments described in the next section of this article implement an evolutionary computation algorithm strongly inspired by the architecture of Canonical genetic algorithm (CGA, P+46) Holland (1992); Rudolph (1994). Hence, initial population is randomly generated and the fitness-proportionate (i.e. "roulette wheel", P+42) selection is used as the main selection operator. But contrary to CGAs, our optimization technique does not implement a classical single-point crossover but rather a sort of "discrete crossover" which takes place only in case that parent individuals have different alleles of a specific gene.

Another reason why our solution can be considered to be more similar to evolutionary strategies (Rechenberg, 1971) than to CGAs is related to the fact that it does not encode individuals as binary vector (P+48). Instead, *every individual represents a candidate mono-alphabetic substitution cipher* application of which could, ideally, transform the Calendar into a crib. More formally: given that cipher is written in symbols of the alphabet $A_{\mathtt{cipher}}$ and given that the crib is written in symbols of the alphabet $A_{\mathtt{crib}}$, then each individual chromosome will have length of $|A_{\mathtt{crib}}|$ genes and every individual gene could encode one among $|A_{\mathtt{cipher}}|$ values.

Size of the search space is therefore $|A_{\mathtt{cipher}}|^{|A_{\mathtt{crib}}|}$. Search for optima in this space is governed by a fitness function:

$$F_{P_{\mathtt{cribbing}}} = \sum_{w \in \mathtt{cipher} \wedge P_{\mathtt{cribbing}}(w) \in \mathtt{crib}} \mathtt{length}(w)$$

where w is a word type occurring in the cipher (i.e. in the Calendar) and which, after being rewritten by $P_{\mathtt{cribbing}}$ also matches a token in the input crib. Given that the expression $\mathtt{length}(w)$ simply denotes w's character length, the fitness function of the candidate transcription procedure $P_{\mathtt{cribbing}}$ is thus nothing else than the sum of char-

Listing 2: Cipher2Dictionary adaptation fitness function

```perl
#Fitness Function
my $text=$calendar;
my $old = "acdefghiklmnopqrsty";
my %translit;
@translit{split //, $old} = split //, $individual;
$text =~ s/(.)/defined($translit{$1}) ? $translit{$1} : $1/eg; #
    core transcription of calendar content
my %matched;
for (split/\n/,$text) {
        my $token=$_;
        if (exists $crib{$token}) {
                @antitranslit{split //, $individual} = split //,
                    $old;
                $token =~ s/(.)/defined($antitranslit{$1}) ?
                    $antitranslit{$1} : $1/eg;
                my $t=$token;
                $matched{$t}=1;
        }
}
for (keys %matched) {
        $Fitness[$i]+=length $_;
}
```

acter lengths of all distinct labels contained in the Calendar which $P_{\mathtt{cribbing}}$ successfully maps onto the feminine names contained in the input crib.

0.5 EXPERIMENTS

Within the scope of this article, we present results of two sets of experiments which essentially differed in the choice of a name-containing cribs.

Other input values (e.g. Takahashi's transliteration of the Calendar used as the cipher) and evolutionary parameters (total population size = 5000, elite population size = 5, gene mutation probability <0.001) were kept constant between all experiments and subexperiments. Each experiment consisted of ten distinct runs. Each run was terminated after 200 generations.

0.5.1 SLAVIC CRIB

What we label as "Slavic crib" is a plain-text list of feminine names which we had compiled from multiple sources publicly available on the Internet. Principal sources of names were websites of western Slavic origin. This choice was motivated by following reasons:

1. The oldest more or less certain trace of VM's trajectory points to the city of Prague - the center of western Slavic culture.

2. Orthography of western Slavic languages relatively faithfully represent the pronunciation. That is, there are relatively few digraphs (e.g. a bi-gram "ch" which denotes a voiced velar fricative). Hence, the distance between the graphemic and the phonemic representations is not so huge as in case of English or french.

3. Slavic languages have rich but regular affective and diminutive morphology which is often used when addressing or denoting beloved persons by their first name.

The third reason is worth to be introduced somewhat further: in both Slavic and western Slavic languages, a simple in-fixing of the unvoiced velar occlusive "k" before the terminal vowel "a" of a feminine names leads to creation of a diminutive form of such a name (e.g. $\mathtt{alena} \rightarrow \mathtt{alenka}, \mathtt{helena} \rightarrow \mathtt{helenka}$ etc.) The fact that this morphological rule is used both by western as well as eastern Slavs indicates that the rule itself can be quite old, date to *common Slavic* or even *pre-Slavic* periods and hence, was quite probably in action already in the period when VM was written.

For the purpose of this article, let's just note that application of the substitution:

$$\mathtt{a\$} \rightarrow \mathtt{ka/}$$

allowed us to significantly increase the extent of the "Slavic crib". Thus, we have obtained a list a of 13815 distinct word types which are in quite close relation to phonetic representation of feminine names used in Europe and beyond[8]. The alphabet of this crib comprises of 38 symbols, hence there exists 19^{39} possible ways how symbols of the Calendar could be replaced by symbols of this crib.

Figure 3 shows the process of convergence from populations of randomly generated chromosomes towards more optimal states. In case of runs averaged in the "SUBSTITUTON" curve, the procedure $P_{\mathtt{cribbing}}$ consisted in simple mapping of the Calendar onto the crib by means of a substitution cipher specified in the chromosome. But in case of runs averaged in the "REVERSAL + SUBSTITUTION" curve, whole process was initiated by the reversal of order of characters present within individual tokens of the Calendar (e.g. $\mathtt{okedy} \rightarrow \mathtt{ydeko}, \mathtt{otedy} \rightarrow \mathtt{ydeto}$ etc.) Let's now look at contents of individuals which were "identified" by the optimization method.

More concrete illustrations can also turn out to be quite illuminating. Hence, if the most elite individual of run 1 (i.e. the one with fitness 197) is as a means of substitution of EVA characters contained in the Calendar, one will see appearance of names like ALENA, ALETHE,

[8] Slavic crib is publicly available at http://wizzion.com/thesis/simulationo/slavic_extended.crib

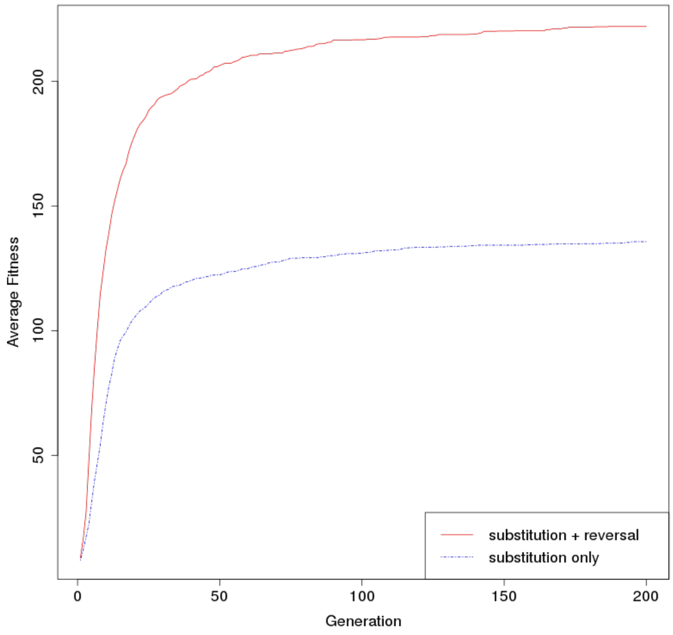

Figure 3: Evolution of individuals adapting label in the Calendar to names listed in the Slavic crib.

ANNA, ATENKA, HANKA, HELENA, LENA etc. And when the last one (i.e. the one with fitness 240 is used), the resulting list shall contain tokens like AELLA, ALANA, ALINA, ANKA, ANISSA, ARIANNKA, ELLINA, IANKA, ILIJA, INNA, LILIJA, LILIKA, LINA, MILANA, MILINA, RANKA, RINA, TINA etc.

This being said, the observation that all reversal-implementing runs have converged to genomes which:

1. transcribe e in EVA as nasal n
2. transcribe k in EVA as velar k
3. transcribe t in EVA as nasal n
4. transcribe y in EVA as vowel a
5. transcribe a in EVA as vowel (80% times as "i", 10% as "e", 10% as "o")
6. transcribe l in EVA as either a liquid consonant (80% "l", 10% "r") or "m" (10%)

...could also be of certain use and importance.

0.5.2 HEBREW CRIB

At this point, a skeptical mind could start to object that what our algorithm adapt to is in fact not the Calendar, but the statistical properties

Fitness	
197	e s t n h k a h k l h t a k a m e n a
230	i k t n s k n h k l z t a j s m i n a
224	i c t n v k / g k l m b a j / r i n a
227	i t n p a f l k l m e a n k r i n a
240	i k t n a k f l k l m e a j g r i n a
226	i l n h o l k r g e a n a m i n a
208	i q g n x k d e k l m x a j x r i n a
239	i k t n d o l l k l f e a k i m i n a
191	o t l n t n n r k m z b a n h r e n a
240	i s t n s k n l k l m e a j I r i n a
EVA	a c d e f g h i k l m n o p q r s t y

Table 1: Fittest chromosomes which map reversed tokens in the Calendar onto names of the Slavic crib

of the crib. And in case of such a long and sometimes somewhat artificial list like Crib_{Slavic}, such an objection would be in great extent justified. For the adaptive tendencies of our evolutionary strategy are indeed so strong that it would indeed find a way to partially adapt the calendar to a crib which is long enough[9]

For this reason, we have decided to target our second experiment not at the biggest possible crib but rather at the oldest possible crib. And given that our first experiment has indicated that it seems to be more plausible to interpret labels in the Calendar as if they were written in reverse, id est from right to left, our interest was gradually attracted by Hebrew language[10]. This lead us to two lists of names:

- $\text{Crib}_{Hebrew-men}$ contains 555 masculine names[11]
- $\text{Crib}_{Hebrew-women}$ contains 283 feminine names[12]

both lists were extracted from the website finejudaica.com/pages/hebrew_names.htm and were chosen because they did not contain any diacritics and

[9] This has been, indeed, shown by multiple micro-experiments which we do not report here due to the lack of space. No matter whether we use cribs as absurd as list of modern American names or Enochian of John Dee and Edward Kelly, we could always observe a sort of adaptation marked by the increase of fitness. But it was never so salient as in case of Crib_{Slavic} or Crib_{Hebrew}.

[10] Other reasons why we decided to focus on Hebrew include: important presence of Jewish diaspora in Prague of Rudolph the 2nd (c.f. the story of rabbi Loew and the Golem of Prague); ritual bathing of Jewish women known as mikveh; usage of VM-resembling triplicated forms (e.g. amen, amen, amen) in Talmudic texts; attested existence of so-called Knaanic language which seems to be principally a Czech language written in Hebrew script et caetera et caetera.

[11] http://wizzion.com/thesis/simulationo/jewish_men

[12] http://wizzion.com/thesis/simulationo/jewish_women

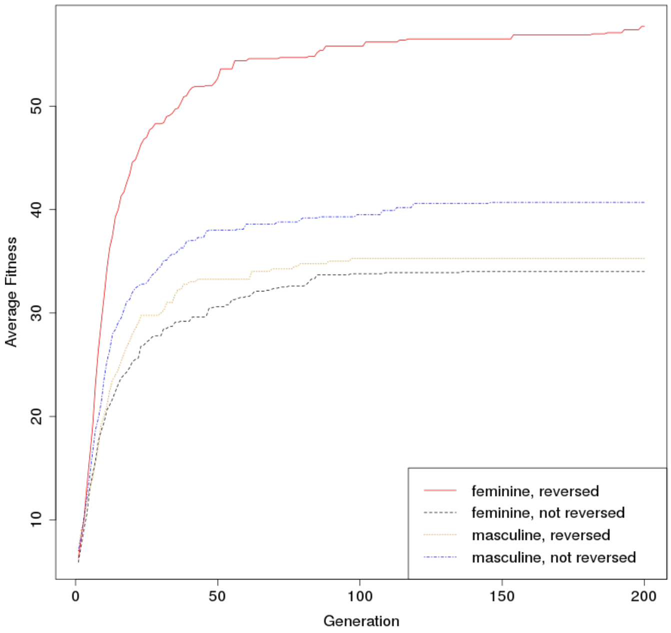

Figure 4: Evolution of individuals adapting label in the Calendar to names listed in the Hebrew cribs.

hence transcribing Hebrew names in a similar way as they had been transcribed millenia ago.

Figure 4 displays the summary of all runs which aimed to transcribe the Calendar with Hebrew names.

As may be seen, the whole system converged to highest fitness values when $\mathrm{Crib}_{\mathrm{Hebrew-women}}$ was used in concordance with reversal of order of characters. In such scenario, minimal attained fitness was attained by run converging to $F_{min(hebrew28,283,hfr)} = 52$, maximal attained fitness was $F_{max(hebrew28,283,hfr)} = 63$. Difference results of hebrew, reverse batch of runs and other results of other batches is statistically significant (Welch Two Sample t-test, p-value $<$ 7e-10).

Subsequently, a list of 283 was tokens randomly generated in a way that the distribution of lenghts of randomly generated sequences is identic to distribution of lenghts of names in the hebrew crib. Maximal attained fitness was $F_{max(random28,283,hfr)} = 26$ among 10 runs aiming to adapt the Calendar to such a random crib. Statistical difference between results of batch of runs adapting to valid character-reversed hebrew crib $hebrew28, 283, hfr$ and the equidistributed randomly generated crib $random28, 283, hfr$ turned out to be strongly significative (Welch two sample two sided non-paired t-test: t = 22.0261, df = 15.442, p-value = 4.384e-13).

The highest attained fitness value was was attained by the cribbing procedure which first reverses the order of characters whose EVA representations are subsequently substituted by a following chromosome:

A	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	Y
'	ה	ד	'	נ	ע	נ	ר	נ	ל	ב	ב	ה	צ	מ	ד	ת	ל	ג

This chromosome transcribes the voynichese Calendar labels *okam, otainy, otey, oty, otaly, okaly, oky, okyd, ched, otald, orara, otal, salal and opalg* to feminine Hebrew names

בינה גברילה גבורה גילה גלה גלילה גלינה גנה דגנה דינה דלילה ידידה לילה לילית עליצה

(i.e. Bina, Gabriela, Ghila, Gala, Galila, Galina, Gina, Degana, Diyna, Deliyla, Yedidya, Lila, Lilit and Alica).

Worth mentioning are also some other phenomena related to these transcriptions. One can observe, for example, that the label "otaly" - translated as Galina - is also present on folios f33v, f34r or f46v which all contain drawings of torch-like plants. This is encouraging because the word "galina" is not only a Hebrew name, but also a substantive meaning "torch". Similarly, the word "lilit" is not only a name but also means "of the night". This word supposedly translates the voynichese token "salal" which is very rare - asides the Calendar it occurs only on purely textual folio f58v and on a folio f67v2 which, surprise!, may well depict circadian rhythms of sunrise, sunset, day and night.

Or it could be pointed out kind that the huge majority of occurrences of voynichese trigram "oky" (potentially denoting the name "gina" which also means "garden") is to be observed on herbal folios. Or the distribution of instances of "okam" (transcribed as "bina" which means "intelligence and wisdom"[13] could, and potentially should, be taken into consideration. Or maybe not.

0.6 conclusion

In 2013, BBC Online had announced "Breakthrough over 600-year-old mystery manuscript". The breakthrough was to be effectuated by Stephan Bax who, in his article, describes the process of deciphering as follows:

« The process can be compared to doing a crossword puzzle: at first we might doubt one possible answer in the crossword, but gradually, as we solve other words around it which serve to confirm letters we have already placed, we gradually gain more confidence in our first answer until eventually we are confident of the solution as a whole.» (Bax, 2014)

What Bax does not add, unfortunately, is that the voynich crossword puzzle is so big that anyone who looks at it close enough can find in it small islands of order, local optima where few characters

[13] Note that "bina" is one among highest sephirots located at north-western corner of kabbalistic tree of life. In this context it is worth noting that only partially readable EVA group "...kam" occurs as a third word near the north-western "rosette" of folio 85v2. Such considerations, however, bring us too far.

seem to fit the global pattern. Thus, even if Bax had succeeded, as he states, in "identification of a set of proper names in the Voynich text, giving a total of ten words made up of fourteen of the Voynich symbols and clusters", this would mean nothing else than that he had identified a locally optimal transcription alphabet.

In this article, we have presented two experiments employing two different lists of feminine names. Both experiments have indicated that if labels in the Zodiac encode feminine names, then these have been originally written from right to left [14]. The first experiment led to identification of multiple substitution alphabets which allow to map 240 EVA letters, contained in 40 distinct words present in the Calendar, onto 35 feminine-name-resembling sequences enumerated among 13815 items of $\mathtt{Crib}_{\mathtt{Slavic}}$. Results of second experiment indicate that if ever the Calendar contains lists of Hebrew names, then these names would be more probably feminine rather than masculine.

This is, as far as we can currently say, all that could be potentially offered as an answer to the question « Can Evolutionary Computation Help us to Crib the Voynich Manuscript?» (Hromada, 2016). Everything else is - without help coming from experts in other disciplines - just a speculation.

0.7 GENERIC CONCLUSION

Looked upon from a superficial point of view, an article presented in this "zeroth analysis" contains nothing else and nothing more than:

1. a very brief description of a particular enigma commonly known as "Voynich Manuscript"

2. introduction of a so-called "primary mapping" hypothesis potentially able to direct any future tentative to decipher the manuscript

3. discussion of inner workings of an "evolutionary algorithm" programmed whose source code is hereby transferred to the public domain[15]

4. presentation of *fairly reasonable* results obtained after confrontation of the manuscript with the algorithm which takes lists of Slavic and Hebrew names at its input

What is meant by the attribute *fairly reasonable* is, of course, a place for argument. And contrary to legions of other researchers, we do not pretend that we have succeeded to "crack" the manuscript. We

[14] Note, however, that this does not necessarily imply that the scribe of VM (him|her)self had written the manuscript in right-to-left fashion. For example, in case (s)he was just reproducing an older source which (s)he didn't understand, his|her hand could trace movements from left to right while the very original had been written from right to left

[15] http://wizzion.com/thesis/simulation0/voynich.PERL

simply state that after being executed on a single core of 1.8GHz CPU, a simple 160-line script written in pure PERL can yield, in just few hours, intelligible transcriptions of "*lattices of terms*" contained in a previously unknown corpus. Thanks to a fairly trivial derivative of a Canonical Genetic Algorithm, an average home PC can closely approximate a brute-force search which would otherwise run weeks (at least) even when executed at state-of-the-art computational clusters.

Simply stated, our 0th simulation indicates that, which has already been indicated many times before:

Evolution narrows-down the search to regions where most plausible hypotheses reside.

Non-negligible speed-up goes hand in hand with such narrowing-down. And it is evident that such speed-up can be useful for any system which can invest only limited amount of time and energy into its search of the most optimal hypothesis. It does not really matter whether the system in which we speak in this context is a PERL script, child's mind or the Nature herself: problem-solving system which implements evolutionary principles tends to converge (Rudolph, 1994) to "the answer" in less time, and with less resources wasted, than the system which does not implement such principles.

At least as fascinating as her ability to speed things up is evolution's propensity to produce adaptations. Zeroth simulation is particularly instructive in this regards: as noted in the footnote 9, the VM-to-crib transcribing EA produced certain results even in cases when cribs as "list of 20th century American names" have been used as target dictionaries. In spite of absurdity of such cribs - for it is indeed highly improbable that VM initially contained names like Butch or Mitch - the EA succeed to discover certain inherent similarities between two texts in order to exploit them in the future search.

Thus, the main conclusion of 0th simulation can be stated as follows:

Evolution is able to facilitate the search for optimal mapping between distinct corpora encoded in distinct forms of representation.

In this simulation, distinct forms of representation has been a so-called EVA alphabet (into which VM is transcribed) and phonemic alphabets common to Slavic or Hebrew languages. Mapping itself was nothing else than simple substitution of one symbol from one alphabet with one symbol from another alphabet. A mapping - a hypothesis - was considered "the fittest" if it succeeded to transcribe initial unintelligible EVA corpus to intelligible list of names. Both EVA corpus and the name list were EA's inputs and thus in certain sense "innate" to each individual run of the algorithm.

What was "acquired" during the process was the set of mono-alphabetic substitution rules. EA presented in 0th simulations is thus an example of evolution which processes strictly "symbolic" representations.

This will not be the case in simulations which are now to follow: let's now descend to the realm of sub-symbolic (vectorial) entities in order to propose an evolutionary solution to the problem of category induction.

0.8 zeroth simulation bibliography

Amancio, D. R., Altmann, E. G., Rybski, D., Oliveira Jr, O. N., and Costa, L. d. F. (2013). Probing the statistical properties of unknown texts: application to the voynich manuscript. *PloS one*, 8(7):e67310.

Bax, S. (2014). A proposed partial decoding of the voynich script. *University of Bedfordshire, http://stephenbax. net/wp-content/uploads/2014/01/Voynich-a-provisionalpartial-decoding-BAX. pdf.*

Champollion, J. F. (1822). *Observations sur l'obelisque Egyptien de l'Ile de Philae*.

Currier, P. (1970). 1976." voynich ms. transcription alphabet; plans for computer studies; transcribed text of herbal a and b material; notes and observations.". *Unpublished communications to John H. Tiltman and M. D'Imperio, Damariscotta, Maine*.

d'Imperio, M. E. (1978). The voynich manuscript: an elegant enigma. Technical report, DTIC Document.

Frege, G. (1994). Über sinn und bedeutung. *Wittgenstein Studien*, 1(1).

Hodgins, G. (2014). Forensic investigations of the voynich ms. In *Voynich 100 Conference www. voynich. nu/mon2012/index. html. Accessed*, volume 4.

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–72.

Hromada, D. (2016). What can evolutionary computation teach us about the voynich manuscript? refused at Cryptologia journal. Refusal review of a single anonymous reviewer accessible at http://wizzion.com/voynich/cryptologia_review.pdf.

Imai, M. and Haryu, E. (2001). Learning proper nouns and common nouns without clues from syntax. *Child development*, 72(3):787–802.

Kennedy, G. and Churchill, R. (2005). *The Voynich manuscript: the unsolved riddle of an extraordinary book which has defied interpretation for centuries*. Orion Publishing Company.

Landini, G. and Zandbergen, R. (1998). A well-kept secret of mediaeval science: The voynich manuscript. *Aesculapius*, 18:77–82.

Newbold, W. R. (1928a). *Cipher of Roger Bacon*. University of Pennsylvania Press.

Newbold, W. R. (1928b). *Cipher of Roger Bacon*.

Rechenberg, I. (1971). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dr.-Ing*. PhD thesis, Thesis, Technical University of Berlin, Department of Process Engineering.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *Neural Networks, IEEE Transactions on*, 5(1):96–101.

Rugg, G. (2004). An elegant hoax? a possible solution to the voynich manuscript. *Cryptologia*, 28(1):31–46.

Schinner, A. (2007). The voynich manuscript: evidence of the hoax hypothesis. *Cryptologia*, 31(2):95–107.

Strong, L. C. (1945). Anthony askham, the author of the voynich manuscript. *Science*, 101(2633):608–609.

Timm, T. (2014). How the voynich manuscript was created. *arXiv preprint arXiv:1407.6639*.

Ventris, M. and Chadwick, J. (1953). Evidence for greek dialect in the mycenaean archives. *The Journal of Hellenic Studies*, 73:84–103.

1 EVOLUTIONARY LOCALIZATION OF SEMANTIC PROTOTYPES

1.1 GENERIC INTRODUCTION

How does a child create mappings between "signifiers" and "signifieds" (de Saussure, 1916), between words and their meanings? How do concepts emerge in the mind of a child?

These question are addressed on many places of Conceptual Foundations. Be it during our discussion of "ontogeny of lexicon and semantics" (P+72-78) or *classical* theories thereof (P+93-95), be it during the definition of "category prototype" (P+132) or in the Hebb/Harris analogy (P+133) suggesting *a sort of equivalence* between Hebb's law well-known to neuroscientists and so-called "distributional hypothesis" well-known to linguists, it has been indicated on multiple places that what contemporary linguists label as "vocabulary development" is, in its essence, nothing else than a usage-based, goal-oriented, associanist process. And that Chomsky's critic of Skinner (P+95), in regards to acquisition of meanings, quite inappropriate: in fact it does not even apply. This is so because first syntactic representations (P+173-179) are acquired, tuned and perfectioned later than first semantic constructions (P+179-184).

And how could such "vocabulary development" be simulated by an engineer willing to do so ?

In an ideal world, such an engineer would have to have, at least, two things at his disposition:

- a corpus C representing the world of a modal toddler: it should contain representations of objects with many attributes (some of them could and should mutually overlap)
- an algorithm A capable of clustering objects into categories in a "cognitively plausible" (P+13) way (i.e. similar to the way child's mind does it)

Unfortunately, as far as 2016, no such C is available, at least not in textual form which could be processed by means of methods commonly used in computational linguistics (P+112-164). The corpus CHILDES (P+207-209) is as close as one can get to C but, and this is a non-negligible "but", CHILDES contains transcripts representing interactions within certain worlds BUT does not contain descriptive representations of these worlds *selves*. And as we have noted elsewhere (Hromada and Gaudiello, 2014) construction of such corpus surpasses by far possibilities of any individual engineer and thus also possibilities of this dissertation.

Willing to develop A but without proper C, one is obliged to approximate. In regards to simulations of induction of meaning, a plausible approximation could be proposed as follows:

Let's suppose that text documents are "objects" and that groups of objects which have similar semantic content (i.e. refer to or speak about similar things) delimit a certain "semantic category".

Under such supposition - and under such supposition only - can one reduce the problem of vocabulary development to a problem of multi-class categorization of documents. Under such *ceteris paribus* - and under such *ceteris paribus* only - can one pretend that the model first published in the article « Genetic Optimization of Semantic Prototypes for Multi-class Document Categorization» (Hromada, 2015) could , in the long run, potentially lead to full-fledged computational models of vocabulary development.

1.2 INTRODUCTION

In computational theories and models learning, one generally works with two types of models: regression and classification. While in regression models one maps continuous input domain onto continuous output range, in models of classification, one aims to find mappings able to project input objects onto a finite set of discrete output categories.

This article introduces a novel means of construction of a particular type of the latter kind of learning models. Due to finite and discrete nature of its output range, classification - also called categorization by more cognition-oriented researchers - seems to be of utmost importance in any cognitively plausible (Hromada, 2014a) model of learning. But under these terms, two distinct meanings are confounded and the term categorization thus often represents both:

1. process of learning (e.g. inducing) of categories

2. process of retrieving information from already learned (induced) categories

which crudely correspond to training, resp. testing phases of supervised learning algorithms.

In the rest of this section we shall more closely introduce an approach combining notions of category prototype, dimensionality reduction and evolutionary computing in order to yield a potentially "cognitively plausible" means of supervised machine learning of a multi-class classifier. We shall subsequently present specificities of a Natural Language Processing (NLP) simulation which was executed in order to assess the feasibility of our approach. Results hence obtained shall be subsequently compared with comparable "deep learning" semantic hashing technique of (Salakhutdinov and Hinton, 2009). The article shall be concluded with few remarks integrating whole research into more generic theories of neural and universal Darwinism.

1.2.1 GEOMETRIZATION OF CATEGORIES

In contemporary cognitive science, categories are often understood as entities embedded in an Δ-dimensional feature space (Gärdenfors, 2004). The most fundamental advantage of such models, whose computer sciences counterparts are so-called "vector symbolic architectures" (VSAs) (Widdows and Cohen, 2014), is their ability to geometrize one's data, i.e. to represent one's data-set in a form which allows to measure distances (similarities) between individual items of the data-set.

Thus, even entities like "word meanings" or "concepts" can be geometrically represented, either as points, vectors or sub-spaces of the enveloping vector space S. One can subsequently measure distances between such representations, e.g. distance of the meaning of the word "dog" from the meaning of "wolf" or "cat" etc. Geometrization of one's data-set once effectuated, space S can be subsequently partitioned into a set R of $|C|$ regions $R = R_1, R_2, ..., R_{|C|}$.

In unsupervised scenario, such partitioning is often done by means of diverse clustering algorithms, the most canonical among which being the k-means algorithm (MacQueen et al., 1967). Such clustering mechanisms often characterize candidate cluster C_X in terms of a geometric centroid of the members of the cluster. Feasibility of a certain partition is subsequently assessed in terms of "internal clustering criteria" which often take into account distances among such centroids.

In the rest of this article, however, we shall aim to computationally implement a supervised learning scenario and instead of working with the notion of category's geometric centroid, our algorithm shall be based upon the notion of category's prototype. The notion of the prototype was introduced into science notably by theory of categorization of Eleanore Rosch which departed from the theoretical postulate that:

> "*the task of category systems is to provide maximum information with the least cognitive effort*" (Rosch, 1999)

In seminal psychological and anthropological studies which have followed, Rosch have realized that people often characterize categories in terms of one of their most salient members. Thus, a prototype of category C_X can be most trivially understood as such a member of C_X which is the most prominent, salient member of C_X. For example "apples" are prototypes of category "fruit" and "roses" are prototypes of category "flowers" in western cultural context.

But studies of Rosch had also suggested another, more mathematical, notion of how prototypes can be formalized and represented. A notion which is based upon the notion of closeness (e.g. "distance") in a certain metric space:

"*items rated more prototypical of the category were more closely related to other members of the category and less closely related to members of other categories than were items rated less prototypical of a category*" (Rosch and Mervis, 1975)

Given that this notion is essentially geometric, the problem of discovery of a set of prototypes can be potentially operationalized as a problem of minimization of a certain fitness function. The fitness function, as well as means how it can be optimized, shall be furnished in section 2. But before doing so, let's first introduce certain computational tricks which allow to reduce the computational cost of such search of the most optimal constellation of prototypes.

1.2.2 RADICAL DIMENSIONALITY REDUCTION

There is potentially an infinite number of ways how a data-set D consisting of $|D|$ documents can be geometrized into a $\Delta-$dimensional space S. In NLP, for example, one often looks for occurrences of diverse words in the documents of the data-set (e.g. corpus). Given that there are $|W|$ distinct words occurring in $|N|$ documents of the corpus, one used to geometrize the corpus by means of a N * M co-occurrence matrix M whose X-th row vector represents the X-th document N_X, Y-th column vector represents the Y-th word W_Y and the element on position $M_{X,Y}$ represents the number of times W_Y occurred in N_X.

Given the sparsity of such co-occurrence matrices as well as for other reasons, such bag-of-words models are more or less abandoned in contemporary NLP practice for sake of more dense representations, whereby the dimensionality of the resulting space, d, is much less than $|W|$, $d \ll |W|$. Renowned methods like Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) set aside because of their computational cost, we shall use the Light Stochastic Binarization (LSB) (Hromada, 2014b) algorithm to perform the most radical dimensionality-reducing geometrization possible.

LSB is an algorithm issued from the family of algorithms based on so-called random projection (RP). Validity and feasibility of all these algorithms, be it Random Indexing (RI, (Sahlgren, 2005)) or Reflective Random Indexing (RRI,(Cohen et al., 2010)) is theoretically founded on a so-called lemma of Johnson-Lindenstrauss, whose corollary states that "*if we project points in a vector space into a randomly selected subspace of sufficiently high dimensionality, the distances between the points are approximately preserved*" (Sahlgren, 2005).

Methods of application of this lemma in concrete NLP scenarios being described in references above, we precise that LSB can be labeled as "most radical" variant of RP-based algorithms because:

- it tends to construct spaces with as small dimensionality as possible (in LSB, $d < 300$; in RI or RRI models, $d > 300$)
- LSB tends to project the data onto binary and not real or complex spaces

It can be, of course, the case that such dimensionality-reduction and binarization can lead to certain decrease of discriminative accuracy of LSB-produced spaces. On the other hand, given that dimensionality reduction and binarization necessary bring about reduction of computational complexity of any subsequent algorithm which could be used to explore the resulting space S, such decrease of accuracy is to be more swiftly counteracted by subsequent optimization. The goal of this study is to explore whether such *post hoc* optimization of classifiers operating within dense, binary, LSB-produced spaces is possible, and whether the combination of the two can be used as a novel means of machine learning.

But before describing in more closer such evolutionary optimizations, let's precise that because of its low-dimensional and binary nature, LSB can also be understood as yielding a sort of "hashing function" aiming to attribute similar hashes to similar documents and different hashes to different documents. In this sense, LSB is similar to approaches like Locality Sensitive Hashing (LSH, Datar et al. (2004)) or Semantic Hashing (SH, Salakhutdinov and Hinton (2009)) often used, or at least presented, as *the* solution of multi-class classification of Big-Data corpora. It is with the results of the latter, "deep-learning" approach, that we shall compare our own results in section 1.5.

1.3 GENETIC LOCALIZATION OF SEMANTIC PROTOTYPES

Let $D = \{d_1, ..., d_{|D|}\}$ be a training data-set consisting of $|D|$ documents to which the training dataset attributes one among $|L|$ corresponding members of set of class labels $L = \{L_1, ..., L_{|L|}\}$.

Let Γ denote a tuple $\Gamma = C_1, ..., C_{|L|}$ whose individual elements are sets containing indices of members of D to which a same label L_l is attributed in the training corpus (e.g. $C_1 = \{3, 4, 5\}$ if training corpus attributes its 1st label only to documents d_3, d_4 and d_5).

Let $H = \{h_1, ..., h_{|D|}\}$ be a set of Δ-dimensional binary vectors attributed to members of D by a hashing function F_H, i.e. $h_X = F_H(d_X)$.

Let S be a Δ-dimensional binary (Hamming) space into which members of H were projected by application of mapping F_H.

Then a classificatory pertinence F_{CP} of the candidate prototype P_K of K-th class ($K \leqslant |C|$) can be calculated as follows:

$$F_{CP}(P_K) = \alpha \sum_{t \in C_K} F_{hd}(h_t, P_K) - \omega \sum_{f \not\subset C_K} F_{hd}(h_f, P_K) \quad (1)$$

whereby P denotes the position of the prototype in S, F_{hd} denotes the Hamming distance [1], h_t denotes the hash "true" document belonging to same class as the prototype, h_f is the vector of the "false" document belonging to some other class of the training corpus and α and ω are weighting parameters.

In simpler terms, an ideal prototype of category C is as close as possible to members of C and as far away as possible from members of other categories.

Given such a definition of an ideal prototype, an ideal |C|-class classifier I can be trained by searching for such a set $P = \{P_1, ..., P_{|L|}\}$ of individual prototypes, which minimize their overall classification pertinence:

$$I = \min \sum_{K=0}^{K=|L|} F_{CP}(P_K) \quad (2)$$

In simpler terms, an ideal |C|-class classifier I is composed of |C| individual prototypes which are as close as possible to documents of their respective categories, and as far away as possible from all other documents.

Equations 1 and 2 taken together, one obtains a fitness function which can be optimized by evolutionary computing algorithms. And given that one explores the prototypical constellations embedded in a binary space, one can use canonical genetic algorithms (CGAs, Goldberg (1990)) for the optimization of the problem of discovery of ideal constellation of most pertinent prototypes. We choose CGAs for three principal reasons:

Primo, we choose CGAs mainly for their property, proven in Rudolph (1994), to converge to global optimum in finite time if ever they are endowed with the best-individual protecting, elitist strategy. Secundo, one can obtain practically useful and exploitable increase in speed simply due to the fact that CGAs are conceived to process binary vectors and do so on CPUs which are essentially built for processing such vectors. Tertio, CGAs offer a canonical, well-defined, "baseline" gateway to much more sophisticated evolutionary computing (EC) techniques and are well understood by both neophytes as well as the most experts of the EC community.

For this reason, we consider as superfluous to describe in closer detail the inner workings of a CGA: instead, references (Goldberg, 1990; Rudolph, 1994) are to be followed and read. Given that the particular values of mutation and cross-over parameters shall be specified

[1] Hamming distance of two binary vectors h_1 and h_2 is the smallest number of bits of h_1 which one has to flip in order to obtain h_2. It is equivalent to a number of non-zero bits in a $XOR(h_1, h_2)$ binary vector.

in the following section, the only thing which in which the reader now needs to be reassured is her correct understanding of the nature of data structures which the algorithm hereby proposed shall implement, in order to encode an individual |C|-class classifier:

Given that equation 1 defines a prototype candidate as a position in Δ-dimensional Hamming space and given that equation 2 stipulates that an ideal |C|-class classifier is to be composed of representations of |C| ideal prototype candidates, the data structure representing an individual solution can be constructed by a simple concatenation of |C| Δ-dimensional vectors. Thus, the individual members of the populations which the CGA shall optimize are, *in essentia*, nothing else than binary strings of length |C|*Δ.

1.4 CORPUS AND TRAINING PARAMETERS

In order to be able to compare the performance of our algorithm with non-optimized LSB and SH, same corpus and dimensionality parameters were chosen as those, which are already reported in the previous studies (Salakhutdinov and Hinton, 2009; Hromada, 2014b). Thus, dimensionality of the resulting binary hashes was Δ=128. Every document of the corpus was hence attributed a 16-byte long hash.

A so-called "20newsgroups" corpus[2] has been used. The corpus contains 18,845 postings taken from the Usenet newsgroup collection divided into training set containing 11,314 postings, 7531 being the testing set ($|D_{training}| = 11313, |D_{testing}| = 7531$). Both training and testing subsets are divided into 20 different newsgroups which correspond each to a distinct topic. Given that every distinct topic represents a distinct category label, $|C| = 20$.

Documents of the corpus were subjected to a very trivial form of pre-processing: documents were split into word-tokens by means of [$\hat{\backslash}$w] separator. Stop-words contained in PERL library Lingua::StopWords were subsequently discarded. 3000 word types with highest "inverse document frequency" value were used as initial terms to which the initial random indexing iteration attributed 4 non-zero values. Hashing function $F_H = LSB(\Delta = 128, Seed = 3, Iterations = 2)$ because there were 2 "reflective" iterations preceding the ultimate stage of "binarization".

Once hashes were attributed to all documents of the corpus, the Hamming space S was considered as constructed and stayed unmodified during all phases of subsequent optimizations and evaluations. As CGA-compliant algorithm, the optimization applied generated the new generation by crossing over two parent solutions chosen by the fitness proportionate (e.g. roulette wheel) selection operator. Each among 2560 (128*20) genes was subsequently mutated (i.e. a correspondent bit was flipped to its opposite value) with probability of

[2] http://qwone.com/ jason/20Newsgroups/

0.1%. Population contain 200 individuals, zeroth generation was randomly generated. Elitist strategy was implemented so that all individuals with equally best fitness survived intact the transition to future generation. Parameters α and ω (e.g. equation 1) used in fitness estimation were both set to 1.

Information concerning the category labels guided the optimization during the training phase. During the testing phase, such information was used only for evaluation purposes. Multiple independent runs were executed and values of precision and recall were averaged among the runs in order to reduce the impact of stochastic factors upon the final results.

1.5 EVALUATION AND RESULTS

Every 250th generation, classificatory accuracy of an individual solution with minimal overall classification pertinence (c.f. equation 2) was evaluated in regards to 7531 documents contained in the testing part of the corpus. Following aspects of classifier's performance were evaluated in order to allow comparison with the results with Precision-Recall curves presented in (Salakhutdinov and Hinton, 2009; Hromada, 2014b):

$$Precision = \frac{\text{Number of retrieved relevant documents}}{\text{Total number of retrieved documents}}$$

$$Recall = \frac{\text{Number of retrieved relevant documents}}{|D_{testing}|}$$

The notion of relevancy is straightforward: an arbitrary document D_T contained in the testing corpus is considered to be relevant to query document D_Q if and only if they were both labeled with the same category label, $L_Q = L_T$.

On the other hand, the correct understanding of what is meant by "retrieved" is the key to correct understanding of the core idea behind the functionality of the algorithm hereby proposed. That is: **the prototypes induced by the CGA optimization are to be used as retrieval filters**.

We precise: given a hash h_Q of a query document d_Q, one can easily identify - among $|C|$ prototypes encoded as components of an quasi-ideal constellation I furnished by the CGA - such a prototype P_N which is nearest to h_Q. Subsequently, each among N documents whose hashes are N nearest neighbors of the prototype P_N, should be considered as retrieved by d_Q. Prototypes discovered during the training phase therefore primarily specify, during the testing phase, which documents are to be considered as retrieved, and which not. For all LSB curves present on Figure 5, the size of such retrieval neighborhood was set to N=2000.

Also, in order to obtain viable precision-recall curves, Radius R=(0, ..., $\Delta = 128$) of the Hamming ball was used as a trade-off parameter.

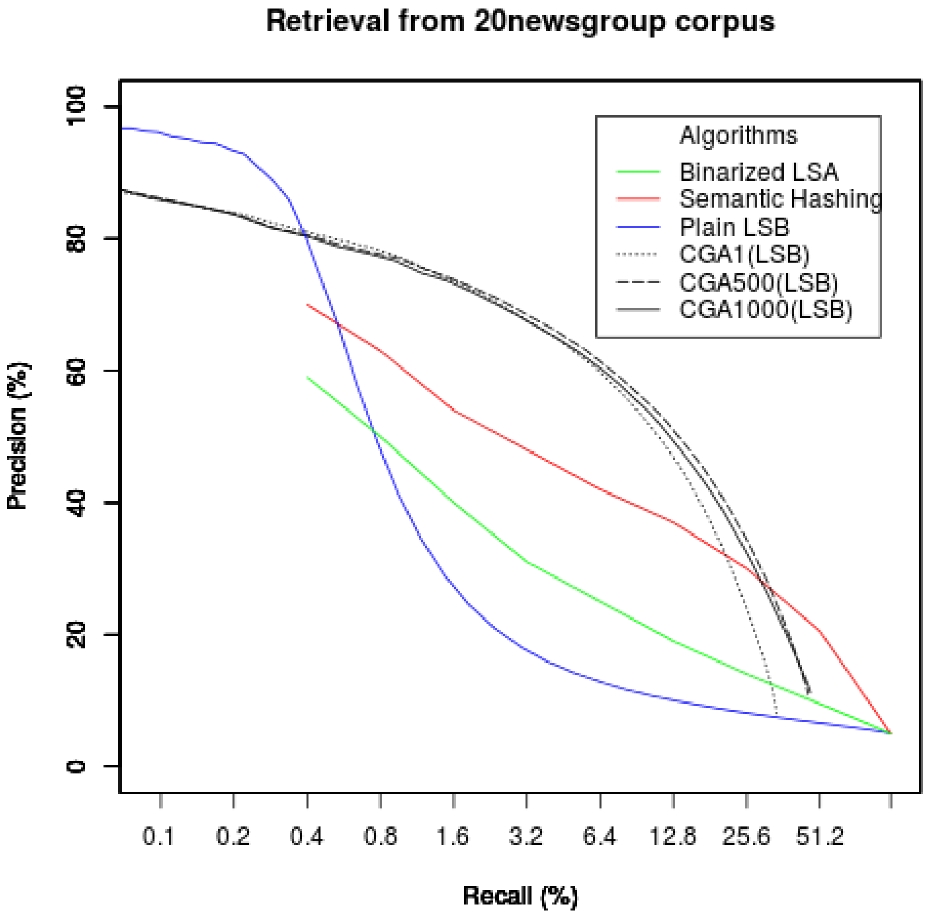

Figure 5: Retrieval and 20-class classification performance in 128-dimensional binary spaces. Non-LSB results are reproduced from Figure 6 of study (Salakhutdinov and Hinton, 2009), plain LSB from (Hromada, 2014b).

For every data-point of the plot on Fig. 1, h_N was considered as retrieved by query h_Q only if the hamming distance of query and the candidate document was smaller than R ($hd(h_Q, h_N) > R$). Points on the very left of the plot correspond thus correspond to R=0 (i.e. h_Q and h_N collide), while points on the right correspond to R=128 (i.e. h_Q does not have a single bit in common with h_N).

As comparison of curves on the figure indicates, *biggest increase in performance is attained by decision to use prototypes as retrieval filters*. Thus, when one uses the most fit among 200 randomly chosen prototype constellations as a retrieval filter (c.f. curve CGA1(LSB)), one obtains significantly better results than when does not use any prototypes at all (c.f. curve "Plain LSB"). If the process is followed by further genetic optimization (c.f. CGA500 for situation after 500 generations), one observes a non-negligible increase of precision in the high recall region of the spectrum. But it can also be seen that the optimization has its limits, hence there is a slight decrease between 500th and 1000th generation which potentially corresponds to situation whereby the induced prototype constellation tends to over-fit the training data-set. This leads to subsequent decrease in overall accuracy of classification of documents contained in the testing data-set.

Figure 5 also suggests that the genetic discovery of sets of prototypes - and their corresponding use as retrieval filters - seems to produce results which are better than those produced by both binarized Latent Semantic Analysis or SH. Exception to this is SH's 20% precision at recall level of 51.2%. Note, however, that since on page 6 of their article, Salakhutdinov and Hinton (2009) claim to have used their hashes as retrieval filters of neighborhood of size N=100, and given that the every size of the category in a 20newsgroup corpus $\approx$ 390 documents, such a result is not even theoretically possible. This is so because even in case the classifying system would retrieve only the relevant documents (i.e. precision would be 100%) the maximal attainable recall would still be just $100/390 \approx 25.6\%$. Both authors were contacted by mail with a request to rectify possible misunderstanding. Unfortunately, none of them replied.

1.6 CONCLUSION

Results hereby presented indicate that supervised localization of constellations of semantic prototypes can significantly increase accuracy of classifiers which use such constellations as retrieval filters.

Given that the localization of such constellations is governed by the training corpus but the increase is also significant in case when one confronts the system with previously unseen testing corpus, we are allowed to state that **our algorithm is capable of generalization**. This was principally attained by combination of following ideas:

1. projection of documents into low-dimensional binary space

2. definition of fitness of prototype in terms of distances to both documents of its category, as well as distance to document of other categories
3. search for fittest prototype constellations
4. use of the most fit prototype constellation as a sort of retrieval filter

In spite of its generalizing and thus "machine learning" capabilities, our algorithms is essentially a non-connectionist one. Thus, instead of introducing synapses between neurons, or speaking about edges between nodes of the graph, briefly, instead of speaking about *deep learning of multi-layer encoders of stacks of Restricted Boltzmann Machines fine-tuned by back-propagation* as (Salakhutdinov and Hinton, 2009) do - we have found as more preferable to reason in geometric and evolutionary terms. It is indeed due to this "geometric" perspective that the computational complexity of the algorithm is fairly low: $\Delta|D||C|$ for evaluation of fitness of one individual prototype constellation. In future study, we aim to explore the performance of slightly modified fitness function whose complexity $\Delta|D| + |C|^2$ could be of particular interest in cases of huge data-sets (i.e. big |D|) with fairly limited number of classes (|C|).

In practical terms, it is also advantageous that both fitness function evaluation as well as final retrieval assess distances in terms of binary hamming distance measure. In both cases, one can use basic logical operations like XOR + some basic assembler instructions which would furnish indices allowing to execute sort of "conceptual geometry" with particular swift and ease. Given these properties + the fact that hashes which are manipulated are fairly small (in one gigabyte of memory, one can store hashes for 8 million documents), one can easily predict existence of future application-specific integrated circuit (ASIC) potentially executing billions query2document comparisons per second.

Computational aspects aside, our primary motive in developing the algorithm hereby proposed was to furnish a sort of cognitively plausible (Hromada, 2014a) "experimental proof" for our doctoral Thesis which postulates that a sort of evolutionary process exists not only in the realm of biological species, but also in realms populated by "species" of a completely different kind.

Id est, in realms of linguistic structures and categories, in realms of word meanings, concepts and, who knows, maybe even in the realm of mind itself.

Being uncertain about whether our demonstrate, with sufficient clarity, that it is reasonable to postulate not only neural (Edelman, 1987), but also intramental evolutionary processes, we conclude by saying that the formula hereby introduced offers a simple yet reasonably accurate method of solving the problem of multi-class categorization of texts.

1.7 GENERIC CONCLUSION

Speaking less concretely, this article shows that model, implementing evolutionary search within a certain type of vector space, can bring practically applicable results. Given that results obtained with training data are in non-negligible extent transposable to testing data, one can consider such model to instantiate a particular case of machine learning (P+125-130). Training data-set is labeled and labels are exploited to direct the evolutionary search: hence, the algorithm can be understood as a supervised one.

Concretely speaking, this article shows how one can perform multi-class (N=20) classification of textual documents. Hence, newspaper articles were considered as entities which are to be classified and occurrence frequencies of words contained within the articles are used as features by means of which the articles are characterized.

And speaking less concretely again, this chapter indicates that evolutionary computation can provide the means to identify constellations of regions in a semantic space which roughly correspond to constellations of semantic categories [3]. Ideally, the process converges to state where correct category labels are attributed to correct regions with correct extension.

It is in this sense that the approach hereby introduced can be, mutatis mutandi, understood as a potential model of vocabulary development within individual child. This is so because the aim of vocabulary ontogeny is analogical: one aspires to attribute correct phonic representations ("words", "signifiers", "labels") to correct regions of the conceptual space. As has been observed by other researchers (P+173) or illustrated by the Borgesian *Ding-Dong Mystery* (P+177-179) such process of attribution *appropriate handel to appropriate vessels* is far from being a monotonic descent to most optimal state.

Rather, the process of acquisition of vocabulary is full of periods where the category is either too exhaustive or too specific, full of small adjustments, detours and returns. It is in this sense that the conjecture *learning of words is an evolutionary process* should be interpreted, and it is in this sense that the aspirations of the algorithm hereby introduced are to be understood.

[3] Note that we use terms "semantic category", "semantic class" or "concept" as synonyms.

1.8 FIRST SIMULATION BIBLIOGRAPHY

Cohen, T., Schvaneveldt, R., and Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240–256.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM.

de Saussure, F. (1916). *Cours de la linguistique generale*.

Edelman, G. M. (1987). *Neural Darwinism: The theory of neuronal group selection*. Basic Books.

Gärdenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT press.

Goldberg, D. E. (1990). Genetic algorithms in search, optimization & machine learning. *Addison-Wesley*.

Hromada, D. D. (2014a). Conditions for cognitive plausibility of computational models of category induction. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 93–105. Springer.

Hromada, D. D. (2014b). Empiric introduction to light stochastic binarization. In *Text, Speech and Dialogue*, pages 37–45. Springer.

Hromada, D. D. (2015). Genetic optimization of semantic prototypes for multiclass document categorization. Awarded "best paper" prize in "Applied Informatics" track of Elitech 2015 conference.

Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA.

Rosch, E. (1999). Principles of categorization. *Concepts: core readings*, pages 189–206.

Rosch, E. and Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *Neural Networks, IEEE Transactions on*, 5(1):96–101.

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5.

Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE.

Widdows, D. and Cohen, T. (2014). Reasoning with vectors: a continuous model for fast robust inference. *Logic Journal of IGPL*.

2 EVOLUTIONARY INDUCTION OF A LIGHTWEIGHT MORPHOSEMANTIC CLASSIFIER

2.1 GENERIC INTRODUCTION

The aim of previous chapter was to show that one can use evolutionary computation to induce sufficiently pertinent semantic categories from a corpus of text documents. Individual text documents were understood as "entities", words present within such documents were understood as their "features" and topics[1] to which diverse documents were attributed were understood as "semantic categories".

Analogies between such process of induction of semantic categories and the process of "vocabulary development", occurring in practically every human being since birth until death, have also been made.

In this chapter we shall explore evolutionary models of induction of yet another type of categories which also play a non-negligible role in human linguistic communication. Id est, induction of grammatical categories. And given that a commonly used definition of a grammatical category (GC) as a *grouping of language units sharing some common feature or function* is very general and vague, this chapter shall focus on particular type of GCs, that of "parts-of-speech" (e.g. "nouns", "verbs", "adjectives" etc.). There are three main "technical" reasons which motivate this choice:

- part-of-speech induction (POS-i, P+135-136) and POS-tagging are well-known NLP problems
- in spite of being well-known, relatively few researchers have proposed evolutionary means to solve these problems (P+137-139)
- certain transcripts within the CHILDES (P+196-222) corpus are tagged with POS-labels

and it is the 3rd reason which is to be understood as the most decisive one in regards to "psycho-linguistic" aims of this dissertation. But the ultimate reason for which we have opted to focus on part-of-speech categories is a theoretical one:

Part-of-speech categories tend to integrate word's semantic content with its grammatical function.

[1] Note the congruence between the fact that the word "topic" is derived from the Greek τόπος which means "place" and the fact that in computational semantics, a topic is literally understood as a "place" within the semantic space

In other terms, the very information that "X belongs to category of nouns" informs the one, who already disposes of a certain notion of what a noun is, that X most probably denotes a thing or a state. And the very information that "Y is a member of category of verbs" suggests that Y most probably denotes a process or an activity. In this regards, the appartenance of the word W to the category C is an irreplaceable clue to not only of W's function and position in the enveloping utterance, but also to W's meaning. This is maybe not so important when the meaning of W is already known, but in case of a language-learning toddler, the ability to recognize that $W \in C$ could significantly reduce her difficulties in solving the problem "*to which components in a recently perceived scene should be a novel W associated?*".

Simply stated: POS-categories can help the child to bootstrap (Karmiloff et al., 2009, pp.111-118) herself into the language.

But how does a child construct such categories in the first place? The aim of the article hereby introduced and recently submitted to journal Computational Linguistics (Hromada, 2016a), is to propose an evolutionary answer.

2.2 INTRODUCTION

What is the essence of linguistic categories, how are such categories represented in human mind and how do such representations develop? Questions which intrigue linguists and philosophers since time immemorial, questions of such elusive nature that any proposal aspiring to answer them have to be, per definitionem, only partial and incomplete.

Such epistemological problems notwithstanding, contemporary computer science tends to offer an instructive answer: categories are classes and classes can be operationalized as regions within an Δ-dimensional vector space S_Δ. Under such definition, training of a categorizing system (i.e a "classifier") can be simulated as a search for the most accurate partitioning of S_Δ. This holds for categories in general and hence it also holds for linguistic categories in particular.

One possible way how such partitioning can be performed is offered by so-called Support Vector Machines (SVM, Cortes and Vapnik (1995)). Basic idea behind SVMs is simple: the algorithm aims to find such a hyper-plane (also called a "decision boundary") which cuts the vector space into two sub-spaces each of which shall ideally contain only data-points attributed to one class. But not only that: given that many such decision boundaries are often possible and identifiable, an SVM tends to identify the one which maximizes the gap (i.e. margin) between data-points themselves and the boundary. Motivation behind such a choice is simple: the more margin is maximized in regards to objects extracted from the training data-set, the more it can be expected that object extracted from a previously unseen "testing data-set" shall be also projected onto the correct side of the boundary. And very often it indeed does: SVMs are able to generalize.

2.2.1 FROM PLANES TO PROTOTYPES

In spite of their theoretical elegance, SVMs - as well as their neural network "perceptron" counterparts - have one important drawback. That is: SVMs and perceptrons look for a "plane" which cuts the space into partitions. But as is illustrated by Figure 7, data-to-be-classified is very often not "linearly separable": a linear decision boundary is nowhere to be found (Minsky and Papert, 1969). In SVM practice, the problem is often solved by applying a certain "kernel function" (Hofmann et al., 2008) which projects the initial data-set onto the space of higher dimensionality where - if the kernel was well chosen - could be the data separated.

While kernel functions have other pleasing mathematical properties [2], they are highly abstract and of significant« mathematical slant» (Hofmann et al., 2008). This, we believe, makes it almost impossible that kernel-based models could ever be labeled as "cognitively plausible" (Hromada, 2014a). In other terms: it is highly improbable that human cognitive and neurolinguistic system would implement as mathematically precise, pure and fragile a machinery as kernels definitely are.

In this article we shall argue that it is in great extent possible to bypass the problem of "linear separability". This is to be attained by focusing one's attention on neighborhoods points $P_A, P_B, ..., P_X$ supposedly representing categories $A, B, ..., X$ instead of focusing it on linear boundaries $B_{AB}, B_{AX}, B_{BX}...$ which supposedly represent the distinction between A and B; A and X etc.

Hence, categories are to be defined in terms of their prototypes (Rosch and Mervis, 1975; Hromada, 2015). Prototypes themselves are points in S tending to satisfy the following condition:

> *An point P_C can be understood as an optimal prototype of a category C if and only if all data-points attributed to C are closer to P than to any other prototype (P_X, P_Y) simultaneously represented within the system.*

In spite of its surface simplicity, the problem posed by this definition of "the optimal prototype" is not an easy one to tackle in a multiclass scenario: the constraint *closer than any other simultaneously represented prototype* substantially complicates the case. If this constraint wasn't present, the problem of identification of "optimal prototypes" would be trivial: prototype would be the centroid of all members of C. But the condition "closer than any other prototype" makes all components of the system mutually dependent on each other. In the end,

[2] The most prominent of which is related to a so-called "kernel trick" which can significantly speed up the classifier-training process.

one is posed in front of the problem somewhat analogical to the famous three-body problem in physics. That is, a problem of which it is well known that it is insolvable by analytic means (Poincaré and Magini, 1899).

2.2.2 FROM PROTOTYPES TO CONSTELLATIONS

This article aims to demonstrate that the problem of discovery of constellations of optimal prototypes can be approximated by a nature-inspired non-connectionist method. In other terms, we shall use a relatively simple evolutionary algorithm in order to "induce" constellations of prototypes which are closer to training data-points to which they should be close and further from training data-points from which they should be far.

Thus, an individual solution contains a position of each component prototype. Every individual has a genome of length $|C|\Delta$ whereby $|C|$ denotes number of distinct classes and Δ is the dimensionality of the space within which the search is performed. As is common to evolutionary algorithms (EAs), these individual solutions are subjected to process replication, selection and variation across multiple generations. Notions of "far" and "close" are implemented directly in the fitness function so that the evolutionary search minimizes the number of incorrectly positioned "nearest prototypes".

Ideally - id est if EAs parameters have been correctly specified and iff the problem of prototype constellation is optimizable at all - the system should converge to such a constellation of prototypes which could accurately classify both testing and training data.

2.2.3 FROM CONSTELLATIONS TO LIGHTWEIGHT CLASSIFIERS

Note that if EAs could discover and optimize such constellations, then these constellations would yield truly "lightweight" classifiers: solution to the $C-$class classification problem of objects in $\Delta-$dimensional space has length $|C| * \Delta$. To be even more radical, let's precise that the search shall operate within binary $\Delta = 64$ spaces which means that position of every data-point as well as a candidate prototype could be defined by exactly 8 bytes. $5-$class classifiers presented in the next sections are described by no more and no less than $5 * 8 = 48$ bytes.

Another reason why these classifiers can be considered as "lightweight" is the nature of features used to project diverse textual tokens into such $64-$dimensional Hamming spaces. Being aware of results issued from our previous empiric simulations (Hromada, 2014a), we have decided to use three features only, i.e.

- suffix of the word W (i.e. last three characters of the word-to-be-categorized)

- suffix of the word $W_L\mathtt{eft}$ (i.e. word immediately preceding W)
- suffix of the word $W_R\mathtt{ight}$ (i.e. word immediately preceding W) are

in order to transform tokens into geometric entities. No other feature has been used during the geometrization phase of the algorithm.

All this in order to propose a nature-inspired model of induction of part-of-speech categories which is, we believe, at least as "minimalist" as Chomsky's "minimalist" program (Chomsky, 1995).

2.3 METHOD

Algorithm presented in this article is very similar to the one presented in (Hromada, 2015). Procedure starts with characterization of training-corpus entities (i.e. "words") in terms of their features (i.e. "suffixes" of W, W_L and W_R) . These features are subsequently used to project all entities into a 64-dimensional Euclidean space $S_{E(64)}$: this component is known as Random Indexing (Sahlgren, 2005). In following steps, whole "space" is reflected so that entities and features "implicitly connected" in the original corpus shall be more pushed to each other than entities and features which are not so connected: this component is known as Reflective Random Indexing (Cohen et al., 2010). At last but not least, all vectors are "binarized" by a simple binary thresholding procedure known as Lightly Stochastic Binarization (Hromada, 2014b). All this steps yield a binary Hamming space $S_{H(64)}$.

Once $S_{H(64)}$ is constructed, one can proceed to localization of most optimal constellations of category prototypes. This is being done by a fairly standard evolutionary algorithm (EA) which is more closely described in 2.3.2.

Most fit solutions obtained after certain number of generations are subsequently confronted with data extracted from the testing corpus in order to assess EA's capability beyond the training set.

2.3.1 CORPUS

This article is conceived as a part of dissertation addressing the possibility of developing evolutionary models of induction of linguistic categories in (and by) human children. This makes the choice of the corpus quite straightforward: the corpus from which we shall aim to extract first linguistic categories is to be contained in Child Language Data Exchange System (CHILDES, (MacWhinney and Snow, 1985)).

However, not all among 30 thousand transcripts contained in CHILDES (Hromada, 2016b) contain part-of-speech labels. Quality of labels also varies: this is no surprise given that some transcripts were manually labeled and/or corrected by multiple annotators while other transcripts have been labeled only by automatic NLP tools (Sagae et al., 2007).

For this reason we have ultimately focused our interest on one particular corpus: Brown's (Brown, 1973) transcriptions of verbal interactions of a girl named Eve. Primo because Brown's work is seminal for whole discipline of developmental psycho-linguistics. Secundo because it is indeed the Eve section of Brown's corpus whose POS-labels have been, according to (Sagae et al., 2007), manually corrected by human annotators.

Classes

According to (Sagae et al., 2007), each token of CHILDES corpus is labeled with one among 31 part-of-speech tags. However, majority of these tags are used only very rarely and/or denote such categories (e.g. AUX for auxiliaries, REL for relativizers or CONJ for conjunctions) of words which encode only little amount or semantic or deontic information.

It is certain that mastery of words belonging to categories like AUX, REL or CONJ play an important role in development of full/fledged adult-like competence. But given that an objective of our dissertation was to elucidate evolutionary computation can simulate bootstrapping of morphosyntactic categories from semantics (and vice versa), we have decided to focus on induction of five classes only. These are enumerated in table 2.3.1.

Class Tag	CHILDES POS tags	Example words
ACTION	v, part, cop	"think", "saying", "is"
SUBSTANCE	n	"cookies", "cow", "ball"
PROPERTY	adj, qn	"better", "blue", "three"
RELATION	prep	"on", "with", "to"
REFERENCE	pro, det, art	"I", "you", "this", "the"

Table 2: Five classes of interest, their corresponding CHILDES part-of-speech tags, some example word types which instantiate them.

What is common to these classes is, that their member words very often denote *visible* and tangible entities, states and processes. Id est, when a child *hears* these words it can be the case that she also perceives their referents by other senses.

Classification of words labeled with tags OTHER than "v", "part", "cop", "n", "adj", "prep", "pro", "art", "det", "qn" has been excluded from the following analysis. Primo,

- because such words do, more often than not, lack easily recognizable visual semantic contents and should not thus be mixed with words which encode such contents

secundo,

- because in ontogeny of a normal child, items belonging to such more abstract classes are mastered later (i.e. after the "toddlerese" (P+17) stage) than words denoting concepts subsumed under five classes listed in (Tomasello, 2009)

tertio,

- because problem of classification of words into 5 classes is, of course, less computationally complex and hence more tractable than problem of classification into 31 classes

and finally,

- it is far from certain whether categories like "auxiliaries" or "relativizers" are represented *per se* within minds of normal verbally communicating humans, or whether such categories are simply abstractions developed by linguists for their own purposes

All these arguments taken together had made us renounce to tentatives to train 31-class POS-classifier and made us focus on training of 5-class classifier only.

Pre-processing

10443 "motherese" utterances have been extracted from twenty transcripts of Brown's Eve corpus. These are very easy to detect because in CHILDES, every utterance is on a separate line and begins with the trigram denoting the locutor of the utterance (in case of mothers, the trigram is MOT). 10443 lines which follow these "motherese" utterances and begin with marker %mor have been also extracted: these are lines which contain manually annotated POS-labels.

Thus, 10443 line-couplets like this:

Listing 3: Motherese utterance from CHILDES corpus + associated morphological tier.

```
eve05.cha:*MOT: that s a duck .
eve05.cha-%mor: pro:dem|that cop|be\&3S art|a n|duck .
```

have been obtained by executing a simple shell command[3]. Lines beginning with MOT and %mor have been subsequently merged by a PERL script enrich_pos.pl[4] which yields output exemplified by the following listing:

Such is the primary data format of this simulation. In this format, each token is characterized on a separate line along with the utterance in which it occurred, as well as with its "gold standard" class-label which was attributed to it by manual annotators. Individual columns

[3] cd Brown/Eve; grep -A3 -P '^MOT' * | grep -P '(MOT | %mor)'

[4] Publicly available at URL http://wizzion.com/thesis/simulation2/enrich_pos.perl

Listing 4: Primary input format of this simulation.

```
that###REFERENCE###train###that s a duck .
s###ACTION###train###that s a duck .
a###QUANTIFIER###train###that s a duck .
duck###SUBSTANCE###train###that s a duck .
```

are separated by separator. The first column denotes the entity itself (the word token), second column contains its class, third column specifies whether the token occurred in a training or testing part of the corpus and the last column contains whole context within which the token entity occurred (i.e. the enveloping utterance).

Let's precise that the training corpus was extracted from first 12 Eve transcripts (i.e. files eve01.cha - eve12.cha) which describe verbal interactions which occurred before Eve attained 2 years of age. Testing corpus, on the other hand, was composed of 8 files (eve13.cha - eve20.cha) transcribed down as Eve was 2 - 2.$\frac{1}{2}$ years old.

The script enrich_pos.pl thus outputs 12453 training corpus tokens and 8746 testing corpus tokens instantiating 972 (training) and 934 (testing) word types. Almost one half (449) of word types occurring in testing corpus does not occur in the training corpus.

2.3.2 algorithm

This is the core of the model. It consists of two major components:

1. "vector space preparation" (VSP): a trivial suffix-extracting filter is used in order to project text from the primary input onto a $64-$dimensional Hamming space
2. "evolutionary optimization": searches S_{H64} for most discriminative constellations of prototypes

Vector Space Preparation

Approach which was used to "geometrize" the primary textual input shares its essential features with that of Random Indexing ((Sahlgren, 2005)) as well as with other Vector Symbolic Architectures (Cohen et al., 2012) based on so-called Random Projection (Hromada, 2013). We describe it elsewhere as follows:

« Given the set of N objects which can be described in terms of F features, to which one initially associates a randomly generated d-dimensional vector, one can obtain d-dimensional vectorial representation of any object X by summing up the vectors associated to all features F_1, F_2 observable within X. **Initial feature vectors are generated in a way that out of** d **elements of vector, only S among them are set to either -1 or 1 value. Other values contain zero.** Since the "seed" parameter S is much smaller than the total number of elements in the vector (d), i.e. S «d, initial feature vectors are very sparse, containing mostly zeroes, with occasional value of -1 or 1.» (Hromada, 2014b).

Section 2.2.3 has already indicated the nature of features which we shall use to initiate the process of geometrization of textual input. We reiterate: we shall characterize every token T with three principal features only:

1. T's own suffix[5]

2. suffix of the token to T's right

3. suffix of the token to T's left

.

Asides this, only two other "lateral features" are used: token T has feature INIT if it is the initial (i.e. first) token of the utterance. Conversely, it is endowed with feature END if it is the last (i.e. terminal) token of the enveloping utterance.

These 3 principal and/or two lateral features are extracted - during the initial phase of VSP - by a following feature-extracting snippet.

Listing 5: PERL code of suffix-feature extractor

```
sub suffix3_featurefilter {
        my @f;
        my @wrdz=split / /,shift; #utterance in 1st parameter
        my $nam = shift; #token of focus in the 2nd
        my ($index)= grep { $wrdz[$_] eq $nam } 0..$#wrdz;
        $index+=1;
        my $pos = 1;
        for my $w (@wrdz) {
                my $w=lc $w;
                my $s=substr $w,-3;
                my $n=$index-$pos;
                $n=$n*-1; #features with minus to the left
                push @f, $n.$s if (abs($n)<2); #main 3 features
                $pos++;
        }
        push @f,"INIT" if $index==1; #lateral feature
        push @f,"END" if $index==scalar(@wrdz); #lateral feature
        return @f;
}
```

For example, when the Random Indexing procedure makes the following call:

suffix3_featurefilter("that s a duck","that")

[5] What we label as suffix SFX_T of token T is, for the purpose of this text, equivalent to T's terminal character trigram (i.e. T's last three letters).

it returns three features characterizing this concrete occurrence (i.e. token) of the word "that":

INIT 0hat 1s

Accordingly, features −1hat, 0s, 1a would be used to characterize this instance of the token s and features −1a, 0uck, END would characterize this instance of duck.

This is the last level of representation which can still be understood as "symbolic". Subsequently, Random Indexing associates a random, sparsely non-zero init vector to each distinct feature (e.g. INIT, END, 0hat, −1hat, 1s, 0s, −1s, −1a, 1a, 0a, 1uck, 0uck etc.) present in any motherese utterance of the Brown/Eve corpus.

All in all, presence of 1321 distinct features has been assessed in the training corpus.

Once features are extracted, things go geometric. Vector representations for individual tokens are obtained as sums of vector representations of associated features. Subsequently, initial random feature vectors are discarded and features themselves are characterized as sums of vector representations of associated tokens. This steps marks the first "reflective" iteration of the process called Reflective Random Indexing (RRI). C.f. Cohen et al. (2010) for closer description of how and why RRI works.

For the purpose of this article, let's just precise that introduction of 2 max 3 "reflective iterations" practically always increases results of one's experiment. This is, in sense, quite expected: for what the reflective process does is not only enriching the representations of entities (e.g. tokens, documents) with information about their features (suffixes, resp. word occurrences) but also enriches representations of features with information about entities within which they occur.

For example, not only should be the word thinking characterized with the feature "ends with suffix −ing" but, conversely, the feature "ing is in part characterized by the fact of occurrence in the word thinking.

Note that all vectors produced by RI and RRI are euclidean. After every "reflection", vectors are normalized so that their unit length is 1. After last such reflection, each real number element of each vector is transformed into a Boolean value by a binary thresholding process known as Light Stochastic Binarization (Hromada, 2014b).

Such binarization is the last step of the vector space preparation. At its end, one obtains a binary vector "hash" tending to have a property common to other convergent[6] hashing methods (Datar et al., 2004; Salakhutdinov and Hinton, 2009):

[6] A hashing function F_H is said to be convergent if similarity between its inputs implies similarity of its outputs. On the other hand, F_H is said to be "divergent" if similarity between inputs does not imply similarity between output hashes. Being of strongly divergent nature, functions like SHA2 or MD5 are not to be confounded with convergent hashing which we discuss here.

Similar inputs tend to have similar hashes.

The moment of attribution of binary hash to each token occurring in the corpus marks the end of the "vector space preparation" phase of the algorithm. In the current model, this VSP occurs only once - at the beginning of simulation and is not repeated.

Evolutionary Optimization

Ensemble of all binary hashes obtained from the corpus yields a hamming space S_H with fairly low dimensionality. This is technically very advantageous since measuring distances can be very swift in such spaces: calculating the hamming distance between two binary strings is definitely[7] less costly than calculating a distance between two real (or even complex) vectors.

The fact that we can measure distances swiftly is crucial for our evolutionary approach for measurement of distances constitutes the very core of the fitness function which is to evaluated for every individual member of every single generation of every single run of the simulation. This is exemplified by the following snippet of PERL pseudo-code.

Listing 6: PERL pseudocode of prototype-inducing fitness function

```
my $fitness=0;
for @individual (@population) {
        for $training_token (@training_tokens) {
                $training_token_hash=$hashes{$training_token};
                $training_token_class=$correct_classes{
                    $training_token};
                $true_prototype_distance=hamming_weight(
                    $training_token_hash XOR $individual[
                    $training_token_class]);
                for $incorrect_prototype ($incorrect_classes{
                    $training_token}) { #the innermost cycle
                        $fitness-- if (hamming_weight(
                            $training_token_hash XOR $individual[
                            $incorrect_prototype]) <=
                            $true_prototype_distance);
                }
        }
}
```

As may be seen that the innermost cycle of the fitness function evaluation contains three operations:

1. XOR between vector of the training object $\vec{o}$ and the vector of "false" prototype $\vec{p_F}$: this yields new vector with true values on those positions where elements of input vectors differ

[7] Or at least on an ordinary transistor-based 21st century Turing machine

2. calculation of hemming weight (i.e. number of non-zero bits) of XOR's result[8]: this is equivalent to hamming distance $Hd(\vec{o}, \vec{p_F})$

3. penalization (decrementation of fitness value) for every incorrect prototype $\vec{p_F}$ which is not further from $\vec{o}$ than o's true prototype p_T, i.e.

$$Hd(\vec{o}, \vec{p_F}) <= Hd(\vec{o}, \vec{p_T}) \tag{3}$$

This concrete instance of prototype-inducing fitness function can be further elucidated by a formula

$$F_{object}(\vec{i}, \vec{o}) = \underset{p_x \neq p_T \wedge Hd(\vec{o}, \vec{p_x}) <= Hd(\vec{o}, \vec{p_T}) \implies p_x \hookrightarrow P_F}{|P_F|} \tag{4}$$

which defines the object-wise fitness $F_{object}(\vec{i}, \vec{o})$ of individual solution $\vec{i}$ in regards to vector representation of the training object $\vec{o}$ as a number (i.e. cardinality of a set) of "false" prototypes $|P_F|$ which are not further from $\vec{t}$ as $\vec{o}$'s corresponding (i.e. "true") prototype $\vec{p_T}$.

Subsequently, an overall fitness of the individual chromosome $\vec{i}$ in regards to each and every object occurring in a training corpus T, is a sum

$$F_{total}(\vec{i}) = -\sum_{o \in T} F_{object}(\vec{i}, \vec{o}) \tag{5}$$

The sum is inverted so that whole function is a maximization one. Under such definition the **maximum fitness value is 0 and corresponds to situation where all training corpus objects are closer to their true prototypes than to any other prototype**.

In theory, it may be the case that multiple global optima of such kind exist. In practice, and in case of many vector spaces, such global optima may not exist at all and fitness of any locally optimal states will have negative value.

Fitness function thus defined, the form of representation of individual solutions is quite straightforward:

An individual solution $\vec{i}$ encodes a **constellation** of all candidate prototypes of $|C|$ categories.

This means that, in regards to every single object $\vec{o}$ present in the training corpus T, $\vec{i}$ shall encode not only "true" prototype $\vec{p_T}$ associated to $\vec{o}$ by the training corpus. It shall also encode all prototypes which are not $\vec{o}$'s true prototypes and which - if ever located closer to

[8] Assembler routine for hamming_weight calculation exploiting the POPCNT instruction implemented (on hardware level) of SSE4.2-compliant CPUs (Suciu et al., 2011) is accessible at URL http://wizzion.com/thesis/simulation2/popcount.asm

$\vec{o}$ than $\vec{p_T}$ - should be evaluated as members of a set of "false positive" P_F.

In practice, individual solution $\vec{i}$ is represented as a vector or an ordered tuple which **concatenates** all its components. Number of possible distinct individuals is

$$2^{\Delta * |C|}$$

where Δ is the dimensionality of the space and $|C|$ is the number of classes. Since in our simulations we have focused on partitioning of 64-dimensional space into five classes ($|C|$=5) there exist potentially $2^{64*5} = 2^{320}$ constellations.

Fitness landscape is thus finite but its complete traversal seems to be impossible to execute in a reasonable amount of time [9].

Two evolutionary heuristics has been deployed in order to explore the landscape:

1. CANONIC: a heuristic strongly reminiscent of Canonical Genetic Algorithms (Goldberg, 1990)

2. MERGE_1: an extension to CANONIC which merges independent runs of CANONIC into one big population and continues the evolution further

In both approaches, every generation starts with fitness evaluation for all individuals in the population. Subsequently, a so-called *2-way tournament selection* operator (Sekaj, 2005) selects members of the mating pool. Size of the mating pool equals the size of population. Members of new generation are obtained from the mating pool as follows: two parents (mother and father) individuals are randomly chosen from the mating pool in order to be subsequently "cut" at a randomly chosen point. Segment before the cut is taken from the mother, segment after the cut is taken from the father and new offspring is obtained. Any gene of offspring's genome can be mutated with 0.2% probability: mutation is equivalent to flipping of a bit. Elitism is not implemented and even the most fit individual can be subjected to decay.

There are thus only two aspects in which CANONIC and MERGE1 differ. One difference is the population size: in CANONIC, populations are fairly small (100 individuals) while MERGE1 implements somewhat bigger ones (1000 individuals).

Both heuristics also differ in the way how their initial population are generated. In CGAs one departs *ex nihilo* and CANONIC heuristics is no exception to this rule: genes present in the gene pool of generation 0 are randomly generated. Things are slightly different

[9] At least on clusters of ordinary transistor-based 21st century Turing machines.

in case of MERGE1 heuristics: MERGE1 is initiated by populations yielded by different runs [10] of CANONIC after 200 generations.

CANONIC and MERGE1 taken together can be thus understood as a very primitive form of "parallel genetic algorithm" (PGA) (Sekaj, 2004).

Under this view, 100 independent runs of CANONIC can be understood as independent nodes on the lower level of the hierarchy and MERGE1 as the node of the higher level. A "migration" from all low-level nodes occurs after 200 generations. Follows a big tournament in which the initial MERGE1 population is constituted.

Subsequently, MERGE1 evolves further.

Parameters

VSP	Input corpus	Brown-Eve motherese [11]
	Feature Filter	suffix3
	Dimensionality	$\Delta = 64$
	Seed	$S = 3$
	Reflections	$I = 3$
CANONIC	Population size	$N = 100$
	Selection	Tournament
	Crossover	One-point
	Mutation rate	$M = 0.2\%$
	Initial population	ex nihilo
	Generations	$G = 200$
	Elitism	$E = 0$
	Runs	$R = 100$
MERGE1	Population size	$N = 1000$
	Selection	Tournament
	Crossover	One-point
	Mutation rate	$M = 0.2\%$
	Initial population	results of CANONIC
	Generations	$G = 300$
	Runs	$R = 6$
Machine Learning	Classes	$\vert C \vert = 5$

Table 3: Parameters of simulation 2.

[10] Note that one common "vector space preparation" phase preceded all CANONIC runs. Hence, in spite of the fact that diverse runs of CANONIC followed different evolutionary trajectories, they always did so in the space S_{64} explored by other runs as well. This makes it possible to "merge" results of different runs.

[11] Available at http://wizzion.com/thesis/simulation2/eve12-8-5classes.mot

2.3.3 evaluation

Accuracy of induced classifiers was primarily evaluated in terms of quantity of correctly predicted category labels (i.e. true positives). Hence, maximum score of 100% would correspond to situation when all objects have been successfully classified. On the contrary, a classifier attributing category membership by random would have precision of cca. 20% in case of classification into 5 equidistributed classes.

Overall classification accuracy of classifiers induced by CANONIC and MERGE1 heuristics has been evaluated after each 10 generations of the training process. Asides this, each class has been explored individually in order to yield class-specific precision and recall values.

Three other classification methods have been evaluated in order to compare the evolutionary method with non-evolutionary approaches:

- CENTROID$_{\text{HAMMING}}$ and CENTROID$_{\text{EUCLIDEAN}}$ baselines
- MSVM (i.e. a Multi-class Support Vector Machine)

Two baseline approaches characterize every class by their centroid. In CENTROID$_{\text{HAMMING}}$ approach is centroid C_X of a category X a hash obtained as an average of hashes of all objects belonging to X. Things are similar in case of CENTROID$_{\text{EUCLIDEAN}}$: the only difference being due to the fact that elements of objects and centroid vectors are now represented in their real-valued form. Id est, a representation issued from the last reflective iteration of the RRI component of the VSP phase of our algorithm.

At last but not least, binary vector space issued from the VSP phase has been partitioned by means of a MSVM implemented in the open-source package MSVMPack (Lauer and Guermeur, 2011). Default settings of the package have been used: linear kernel has been applied and training of MSVM2 (Guermeur and Monfrini, 2011) model has been stopped after converging to 98% accuracy level.

2.4 discussion of results

Table 4 summarizes main results of five compared methods. Smallest amount of correctly classified tokens was attained by baseline CENTROID approaches: this was expected since these approaches do not include any optimization at all[12]. The observation that CENTROID$_{\text{HAMMING}}$ is less precise than CENTROID$_{\text{EUCLIDEAN}}$ is also trivial: transformation of real-valued vectors into binary ones brings about a non-negligible information loss. Worse performance of binary-based classifiers is a result of this information loss.

Optimization, however, can significantly reduce or even counteract impact of such loss. Hence, even a fairly simple CANONIC genetic

[12] Note, however, that classification accuracy of these models is still significantly superior to a random classifier.

Method	Training corpus	Testing corpus
CENTROID$_{\text{HAMMING}}$	455 (42.12%)	412 (40.47%)
CENTROID$_{\text{EUCLIDEAN}}$	572 (52.96%)	533 (52.35%)
MEAN(GA$_{\text{CANONIC}}$)	631 (58.44%)	589 (57.88%)
MEAN(GA$_{\text{MERGE1}}$)	718 (66.51%)	657 (64.57%)
FITTEST(GA$_{\text{MERGE1}}$)	772 (71.48%)	699 (68.66%)
MSVM2	781 (72.31%)	736 (72.30%)

Table 4: Overall results of five different approaches. GA results have been averaged across diverse runs (R = 6*100 for CANONIC, R=6 for MERGE1).

algorithm discovers, in just five sweeps through the hamming space, constellations of prototypes whose precision is higher than that of Euclidean centroids. This is exemplified by Figure 6 which plots evolution of precision across generations.

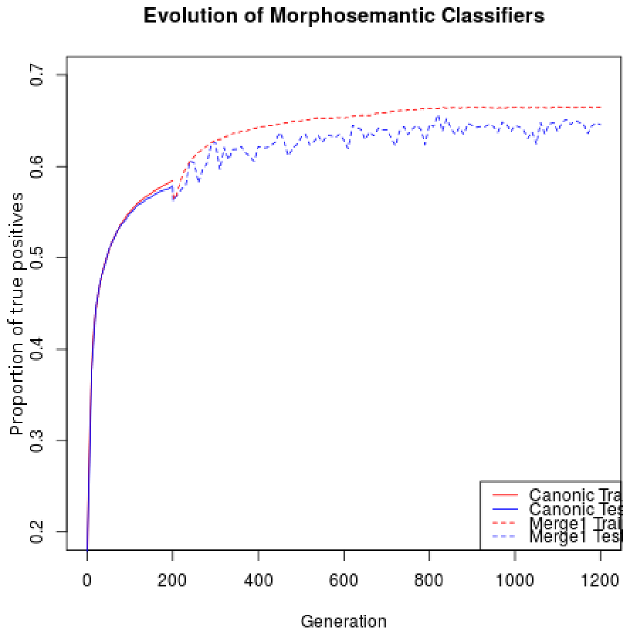

Figure 6: Evolutionary optimization increases the precision of a multi-class classifier. Curves represent results averaged across diverse runs (R = 6*100 for CANONIC, R=6 for MERGE1)).

It may be seen that introduction of PGA-like approach - as exemplified by MERGE1 - results in a significant increase in amount of precisely classified tokens. The score is still not so high as that of MSVM2 (compare 781 with 718 for training corpus, resp. 736 with 657 in testing corpus), but the jump between CANONIC and MERGE1 suggests that that another PGA architecture, introduction of elitism or a different choice of parameters or operators can potentially result in significant boost.

Table 5: MSVM2 training corpus confusion matrix.

	ACT	SUB	PROP	REL	REF
ACT	266	54	0	0	1
SUB	55	495	4	0	0
PROP	21	66	18	0	0
REL	20	12	1	2	0
REF	15	47	3	0	0

Table 6: MSVM2 testing corpus confusion matrix.

	ACT	SUB	PROP	REL	REF
ACT	271	38	4	0	1
SUB	55	450	8	1	0
PROP	21	62	15	0	0
REL	20	6	3	0	0
REF	20	38	5	0	0

Table 7: Training corpus confusion matrix produced by FITTEST(GA$_{\text{MERGE1}}$).

	ACT	SUB	PROP	REL	REF
ACT	278	28	4	4	7
SUB	56	427	34	18	19
PRO	19	39	43	3	1
REL	15	5	1	11	3
REF	9	35	7	1	13

Table 8: Testing corpus confusion matrix produced by FITTEST(GA$_{\text{MERGE1}}$).

	ACT	SUB	PROP	REL	REF
ACT	269	21	9	6	9
SUB	62	371	41	25	15
PRO	16	35	40	3	4
REL	15	3	4	5	2
REF	11	26	8	4	14

As may be seen on confusion matrices shown on tables 5 - 6, MSVM2 fails to correctly classify any testing corpus token attributed to minor REL and REF categories (i.e. recall = 0%) and the situation is not better in case of PROP class neither (testing recall 15.3%)[13]. On the other hand, this handicap is counteracted by MSVM's higher recall rates in regards to dominating SUB and ACT classes. This could potentially suggest that MSVM still tends to behave like a good old "dualist" Support Vector Machine rather than a truly multi-class classifier.

Confusion matrices on tables 7-8 indicate that FITTEST(GA$_{\text{MERGE1}}$) also performs quite well when it comes to classification of tokens into major categories ACTION (86.6% recall; 73.74% precision) and SUBSTANCE (73.74% recall; 79.96%precision). Asides this, it also attains 40% testing recall for PROPERTY class and 22% testing recall for the REFERENCE class. This suggests that even categories of minor importance play a certain role in models induced by evolutionary search for prototype constellations.

[13] These low recall rates imply that the average F1 score of MSVM is, in fact, inferior that of FITTEST(GA$_{\text{MERGE1}}$). This is the case for both training ($F_{\text{MSVM}} = 0.481; F_{\text{FITTEST(MERGE1)}} = 0.518$) as well as testing ($F_{\text{MSVM}} = 0.426; F_{\text{FITTEST(MERGE1)}} = 0.474$) phases.

2.5 conclusions

2.5.1 Computational conclusion

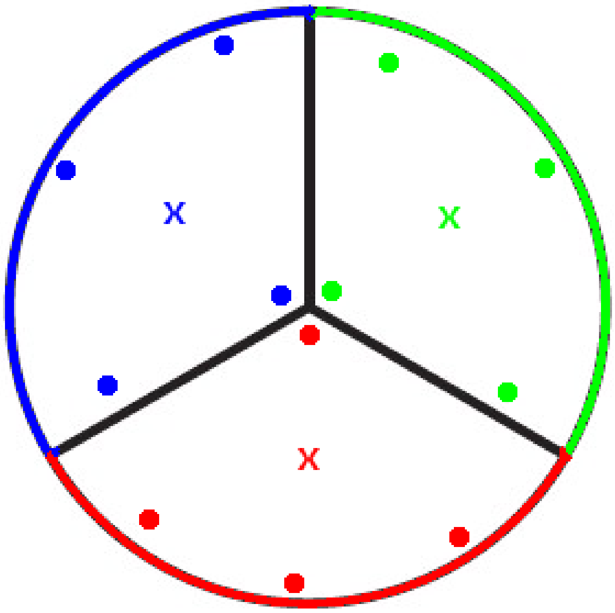

Figure 7: Centroidal tessellation of twelve data-points belonging to three distinct classes. Dots represent data-points, crosses are category prototypes and colors denote category membership. Black lines denote tesselation boundaries.

Figure 7 displays a potential training data-set composed of twelve data-points attributed to three distinct classes. One can observe that it is not possible to draw a single straight line which would separate all datapoints of one class from data-points of other classes. Hence, these data-points are plainly not separable by a linear boundary: many a researcher would be tempted to say that in order to classify such dataset, one would be obliged to apply a certain kernel and project it into space with higher dimensionality.

This is, however, not necessary, if one applies a machine learning strategy which looks for constellation of points instead of lines, planes or hyper-planes. Denoted on Figure 7 by crosses of different colors, such points - labeled as "category prototypes" - satisfy one simple condition:

Every data-point is closer to its prototype than to any other prototype.

Search for constellations of prototypes which satisfy such condition can be thus understood as a problem closely related to problems of Voronoi-Dirichlet tessellations (Aurenhammer, 1991). But contrary to such approaches where "seed" "generator" points are given in advance, positions of such points are, in our approach, induced by means of evolutionary computation.

Inductive process described in this article took place in a 64-dimensional binary space. Reasons behind this choice were of pragmatic nature:

1. optimization involving calculation of Hamming distances can be very fast, especially when implemented on arrays of dedicated Field Programmable Gate Arrays (Sklyarov and Skliarova, 2014) or Application Specific Integrated Circuits

2. binary hashes are very concise form of representation: our approach could be thus useful in Big Data scenarios[14]

These reasons aside, nothing forbids to bypass the "binarization procedure" and search for constellations of prototypes in an Euclidean space. It can be expected that precision of classifiers induced in Euclidean space would be higher than precision of classifiers induced in binary spaces. However, since there is no free lunch, such euclidean search would be undoubtedly more demanding when it comes to consumption of both memory and computational resources.

Shortcomings related to decision to execute the search in binary space notwithstanding, obtained results are quite encouraging. Hence, in a scenario aiming to classify tokens occurring in Brown-Eve section of the CHILDES corpus into 5 morphosemantic classes, classifiers induced by evolutionary optimization identified almost as many true positives as a multi-class SVMs (Lauer and Guermeur, 2011). In terms of F1-Score obtained as a harmonic mean of average recall and average precision, the performance of the most fit prototype constellation FITTEST(GA$_{\text{MERGE1}}$) turned out to be even higher than that of MSVM2. This, however, is more a residuum of a F1-score metrics than a result which would merit to be reported elsewhere than in a fotnote$_{13}$.

2.5.2 psycholinguistic conclusion

Table 9 lists tokens located in closest neighborhoods of three major prototypes which have been encoded in the constellation FITTEST(GA$_{\text{MERGE1}}$).

A subsequent inspection of false positives present in Table 9 turns out to be quite instructive. Hence, the token "building", present in the utterance "what are you building here?" on line 5417 of eve05.cha transcript is clearly not a noun, as CHILDES annotators supposed, but rather a participle - and hence an instance belonging to ACTION class, as correctly predicted by FITTEST(GA$_{\text{MERGE1}}$). Idem for "hit" present in the utterance "did you hit your head?" present on line 4145 of eve01.cha transcript: the token is clearly not a noun, as postulated

[14] In case of 64-bit hashes one could potentially need as little as 800 Megabytes of storage volume in order to store hash representations of 100 million documents.

P_{ACTION}			$P_{\text{SUBSTANCE}}$			P_{PROPERTY}		
H	TOKEN	POS	H	TOKEN	POS	H	TOKEN	POS
10	pointing	part	10	penny	noun	18	**whistle**	noun
10	tripped	v	10	tummy	noun	20	**bent**	v
11	slipped	v	11	cracker	noun	21	**graham**	noun
11	squashing	part	11	graham+cracker	noun	21	**part**	noun
12	**building**	noun	11	key	noun	21	tough	adj
12	burped	v	11	**matter**	v	22	alright	adj
12	cutting	part	11	nap	noun	22	other	adj
12	dripping	part	11	paddle	noun	22	**pitcher**	noun
12	fixed	v	12	drinker	noun	22	**sweetheart**	noun
12	mix	v	12	letter	noun	23	**a**	art
13	dropped	v	12	**numbers**	v	23	**cough**	noun
13	hit	v	12	paper	noun	23	**fun**	noun
13	**hit**	n	12	snowman	noun	23	**grannie_hart**	noun
13	playing	part	12	**worse**	adj	23	**lemon**	noun
13	are	v	13	bx	noun	23	little	adj
13	saw	v	13	face	noun	23	**through**	prep
13	standing	part	13	maam	noun	24	all_gone	adj
13	swim	v	13	purple	noun	24	good	adj
13	want	v	13	soup	noun	24	bigger	adj
13	wiped	v	13	stove	noun	24	busy	adj

Table 9: Testing corpus tokens closest to prototypes of ACTION, SUBSTANCE and PROPERTY encoded in FITTEST(GA_{MERGE1}) constellation. Hamming distance H(token, prototype) and token's CHILDES part-of-speech annotations. False positives are marked by bold font.

by CHILDES annotators, but, as predicted, a verb and hence member of ACTION class. And one can continue: the token "matter" annotated on lines 2152 and 5688 of CHILDES corpus as a verb is clearly not a verb but a noun - and hence a member of a class SUBSTANCE - because it twice occurs in the utterance "what's the matter?. And in spite of the fact that CHILDES labels the token "numbers" as a verb, it is definitely not a verb when it occurs in the utterance "the numbers are going around too" (eve15.cha, line 6276). Et caetera et caetera.

Thus, in spite of the fact that POS tokens in Brown/Eve section of the CHILDES corpus are supposedly « annotated with high accuracy» (Sagae et al., 2007) it is, sometimes, not really the case. In this regards, one would be tempted to state that, as of 2016 AD, is the frontier between developmental and computational psycholinguistics still resembling a structure standing on clay feet. This is a first conclusion which could be potentially useful to any (comp | dev)psycholinguists willing to undertake the path initiated by the study hereby introduced.

The fact that our approach has allowed us to identify errors in the corpus which even humans didn't succeed to identify, is indeed encouraging. And it is moreso encouraging when one realizes how simple was a feature set which has been used to construct the vector space in which all subsequent classifications took place. We repeat:

Every token T was primarily characterized by:

1. *T's three last characters*
2. *three last characters of the token which precedes T*
3. *three last characters of the token which follows T*

asides this, only other information taken into account concerned T's potential position at the very beginning or end of the utterance.

Reason to depart from such a restricted feature set has been in part empiric (Hromada, 2014a). But there exist others, more profound reasons why we have initiated the *training of a verbally interacting computational agent* with focus on suffix-like features. Primo: the "less is more" hypothesis whose implication for neural-network-based processing of natural language has been so beautifully demonstrated by Elman (1993).

Secundo, note Slobin's operating principle A:

« Pay attention to the ends of words.» (Slobin, 1973)

which, according to its author, is a "general developmental universal".

In this regards does our analysis indeed demonstrate that "ends of words" offer features strong enough to initiate a supervised process of induction of categories which have been, for the purpose of this article, labeled as "morphosemantic".

And that the whole process can yield fruit even when representing a 5-class classifier with representation as concise, as 40-bytes long vector definitely is.

2.6 generic discussion

This chapter has presented an algorithm which succeeds to correctly classify a significant amount of tokens into so-called "morphosemantic classes" (MS-classes). But why should one speak about such MS-classes instead of staying faithful to well-established term "parts of speech" ?

An answer is simple: because MS-classes are sometimes not equivalent to parts-of-speech categories. For example, an MS-category labeled as "ACTION" includes not only verbs, but also participles. Motivation behind this distinction is quite simple: it may potentially make sense for an expert linguist to state that "eating" functions is a participle but "to eat" a verb. However, a modal toddler of 20 months shall most probably turn out be ignorant of such a distinction (Tomasello, 2009). For what counts for such a toddler is the fact that he can associate both words "eat" and "eating" with the fact of simultaneously observing certain invariant structural property of her[15] surrounding environment (i.e. observes *activity* of putting something into one's mouth).

Table 2.3.1 introduced five initial MS-classes[16]. These MS-categories have been defined very loosely in the limited scope of this study: all substantives where defined as belonging to the class SUBSTANCE, diverse verbal, participial and infinitival forms as those instantiating ACTION, adjectives and numerals were collapsed into the MS-class PROPERTY, everything which had something to do with pointing, specification and deictics was subsumed under REFERENCE and prepositions were told to instantiate notion of RELATION.

Said in more practical terms: introduction of the notion of MS-class allowed us to enrich certain section of the CHILDES corpus (i.e. Brown's 20 transcripts of a girl named Eve) (Brown, 1973) with certain amount of *loosely semantic* information. *Loosely* because MS-classes, as used in this chapter, are loosely constructed themselves. For it is not always true that $POS_{substantives}$ always denote substances and POS_{verba} always denote actions: no serious linguist could defend such a general view in more than one article and still stay unostracized by the linguistic community.

Loosely, but in regards to "motherese" addressed to a modal toddler (P+17), also *semantic*. For what is more vital for a 18-month old child, to understand&express the difference between verb "eat" and participle/property "eating", or rather understand&express the difference between act of eating and the object being eaten ?

We summarize: act of making a notational turn from the concept of "parts-of-speech" to the notion of "morphosemantic class" led to enrichment of CHILDES corpus with few bits of semantic information. Few bits maybe, but still more bits than noise. Subsequent coupling of this information with morphological information contained in suffixes followed by optimization by means of an evolutionary algorithm allowed us to converge to very concise, 40-byte long multiclass classi-

[15] To stay consistent with Conceptual Foundations as well as with other books of psycholinguistic tradition, we refer to toddlers and children with feminine pronouns "she", "her" etc.

[16] We leave to reader's own ingenuity the exploration of an extent in which could these MS-classes correspond to Aristotle's categories, or Kant's and Piaget's "forms of pure reason".

fiers. These classifiers have subsequently resulted in identification of errors produced by much more complex and - so the authors pretend- also « highly accurate» (Sagae et al., 2007) POS-tagging systems supposedly corrected by multiple human annotators.

These considerations make us believe that a notion of morphosemantic classifier could be of certain use and applicability for any present or future researcher aiming to deploy, develop or fine-tune certain nature-inspired yet cognitively plausible (Hromada, 2014a) models of ontogeny of linguistic categories [17].

2.7 SECOND SIMULATION BIBLIOGRAPHY

Aurenhammer, F. (1991). Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405.

Brown, R. (1973). *A first language: The early stages.* Harvard U. Press.

Chomsky, N. (1995). *The minimalist program*, volume 28. Cambridge Univ Press.

Cohen, T., Widdows, D., Schvaneveldt, R. W., Davies, P., and Rindflesch, T. C. (2012). Discovering discovery patterns with predication-based semantic indexing. *Journal of biomedical informatics*, 45(6):1049–1065.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Guermeur, Y. and Monfrini, E. (2011). A quadratic loss multi-class svm for which a radius–margin bound applies. *Informatica*, 22(1):73–96.

Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220.

Hromada, D. D. (2013). Random projection and geometrization of string distance metrics. In *RANLP*, pages 79–85.

[17] Proof-of-concept source code of this simulation is freely available at URL http://wizzion.com/thesis/simulation2/ELLA.tgz under mrGPL licence.

Hromada, D. D. (2014a). Comparative study concerning the role of surface morphological features in the induction of part-of-speech categories. In *Text, Speech and Dialogue*, pages 46–52. Springer.

Hromada, D. D. (2014b). Conditions for cognitive plausibility of computational models of category induction. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 93–105. Springer.

Hromada, D. D. (2014c). Empiric introduction to light stochastic binarization. In *Text, Speech and Dialogue*, pages 37–45. Springer.

Hromada, D. D. (2015). Genetic optimization of semantic prototypes for multiclass document categorization. Awarded "best paper" prize in "Applied Informatics" track of Elitech 2015 conference.

Hromada, D. D. (2016a). Evolutionary induction of a lightweight morphosemantic classifier. submitted to Computational Linguistics.

Hromada, D. D. (2016b). Reproducible identification of pragmatic universalia in childes transcripts. Accepted for JADT2016 conference.

Karmiloff, K., Karmiloff-Smith, A., and Karmiloff, K. (2009). *Pathways to language: From fetus to adolescent*. Harvard University Press.

Lauer, F. and Guermeur, Y. (2011). Msvmpack: a multi-class support vector machine package. *The Journal of Machine Learning Research*, 12:2293–2296.

MacWhinney, B. and Snow, C. (1985). The child language data exchange system. *Journal of child language*, 12(02):271–295.

Minsky, M. and Papert, S. (1969). Perceptrons.

Poincaré, H. and Magini, R. (1899). Les méthodes nouvelles de la mécanique céleste. *Il Nuovo Cimento (1895-1900)*, 10(1):128–130.

Rosch, E. and Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., and Wintner, S. (2007). High-accuracy annotation and parsing of childes transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 25–32. Association for Computational Linguistics.

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5.

Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

Sekaj, I. (2004). Robust parallel genetic algorithms with re-initialisation. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 411–419. Springer.

Sekaj, I. (2005). *Evolučné vỳpočty a ich využitie v praxi*. Iris.

Sklyarov, V. and Skliarova, I. (2014). Hamming weight counters and comparators based on embedded dsp blocks for implementation in fpga. *Advances in Electrical and Computer Engineering*, 14(2):63–68.

Slobin, D. I. (1973). Cognitive prerequisites for the development of grammar. *Studies of child language development*, 1:75–208.

Suciu, A., Cobarzan, P., and Marton, K. (2011). The never ending problem of counting bits efficiently. In *Roedunet International Conference (RoEduNet), 2011 10th*, pages 1–4. IEEE.

Tomasello, M. (2009). *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press.

3

EVOLUTIONARY INDUCTION OF 4-SCHEMA MICROGRAMMARS FROM CHILDES CORPORA

3.1 GENERAL INTRODUCTION

First simulation has indicated that one can use evolutionary computation in order to partition a semantic feature space into regions which roughly correspond to certain "topics". Second simulation has shown how an evolutionary search succeeds to increase the accuracy of so-called morphosemantic classifiers. Both simulations differed in regards to corpus-which-was analyzed (20 newsgroups corpus in simulation 1, CHILDES/Brown/Eve corpus in simulation 2) as well as in a feature set used to project initial text into binary vector space.

However, both simulations:

1. were optimized by means of an evolutionary algorithm
2. succeed to transpose knowledge present in the training set in order to correctly classify the elements of the testing set (i.e. generalization)
3. used labeled corpus as input of the learning process

Taken together, points two and three indicate that simulations 1 and 2 can be understood as particular instances of *supervised* machine learning. That is, a case of learning which demands more than exposition to the plain input corpus. In case of supervised learning, one needs to have another, parallel, source of information as well. Category labels which have been manually attributed by human annotators are most common cases of such "parallel" source of information.

It may be the case, however, that certain problems do not necessitate the exposure to such additional input at all. Such is, according to some linguists, also the problem of grammar induction (P+148-162) whereby one aims to infer a grammar of a language L solely from the corpus of utterances of L.

Because of this, computational models of GI are considered to be particular cases of *unsupervised* machine learning[1].

This chapter shall aim to present one particular model of GI. That is, an evolutionary model strongly resembling models presented in

[1] Note, however, that the very act of choosing, in the moment T_0 (and not in T_1) and input corpus C_X and not C_Y can also be considered as an act of supervision. C.f. (Hromada, 2014a, 2016) for further discussion of the "unsupervised" vs. "semi-supervised" dilemma.

previous chapters. But also a model aspiring to induce certain generic "micrograms" from nothing else than the Brown/Eve section of the CHILDES corpus.

Article presented in this chapter has been submitted to journal Evolutionary Computation (Hromada, 2016a).

3.2 INTRODUCTION

Input of Grammar Induction (GI) process is a corpus of sentences written in language L, its output is, ideally a grammar (P+117-P+124) or a transparent language model able to generate sentences of L, including such sentences that were not present in the initial training corpus.

In spite of a seemingly simple nature of the problem, induction of grammars from natural language is quite a difficult nut to crack. Thus, symbolic models like the Syntagmatic-Paradigmatic GI (Wolff, 1988), graph-based ADIOS (Solan et al., 2005; Brodsky et al., 2007) do, indeed, attain interesting results in their efforts to extract English grammar from English corpora.

But given the deterministic nature of these models, they tend to converge to certain local optima from which there is no way out. To make things worse, such models often do not dispose of means which would allow them to purge themselves from unwanted overregularizations (P+83).

In this chapter, we shall present a GI model aiming to harness evolution's ability to *discard the unwanted*. What's more, we shall exploit the genotype - phenotype distinction (Fogel, 1995) in order to perform sub-symbolic variation of sets of symbolic sequences. By doing so, we shall obtain a models which integrates entities represented at two levels of abstraction:

1. sub-symbolic feature vector spaces
2. symbolic PERL-compatible regular expressions

Ideally, such a model could be both robust as well as flexible enough to find its middle path between grammars which cover just one thing, and grammars which cover everything.

3.2.1 TWO EXTREMES

The nature of resulting grammar is closely associated to the content of the initial corpus as well as to the nature of the inductive (learning) process. According to their «expressive power», all grammars can be located somewhere on a «specificity – generality» spectrum. On one extreme lies the grammar having following production rules:

$$1 \rightarrow 2*$$

$$2 \rightarrow a|b|c \ldots Z$$

whereby $*$ means «repeat as many times as You Want» and | denotes disjunction.

This very compact grammar can potentially generate any text of any size and as such is very general. But exactly because it can accept any alphabetic sequence and thus does not have any «discriminatory power» whatsoever, is such a grammar completely useless as an explication of system of any natural language.

On the other extreme of the spectrum lies a completely specific grammar which has just one rule:

$$1 \rightarrow < \mathsf{Corpus} >$$

This grammar contains exactly what Corpus contains and is therefore not compact at all (in fact, it is even two symbols longer than Corpus). Such a grammar is not able to encode anything else than the sequence which was literally encoded by the training Corpus.

Such grammar is therefore completely useless for any scenario were novel sequences are to be generated (or accepted).

The objective of GI process is to discover, departing solely from Corpus (written in language L), a grammar which is neither too specific, nor too general. If it is too general, it shall «over-regularize» (P+83). That is: such G shall be able to generate (or accept) sentences which the common speaker of L would never ever consider as grammatical.

On the other hand, if G is too specific, it shan't be able to represent all sentences contained in Corpus or, if it shall, it shan't be able to generate (or accept) any sentence which is considered to be sentence of L but was not present in the initial training corpus Corpus.

3.2.2 DEFINITIONS

G-Category (DEF)

Let's have a set of N objects $(O_1, O_2, ..., O_N)$ embedded within a Δ-dimensional space S (i.e. every object O_X can be described by a vector $\vec{o}_X = V_1, V_2, ..., V_\Delta$). Then geometrized category (G_Δ-Category) **C is defined as a content of S-embedded D-dimensional sphere** with

1. centroid whose coordinates are given by a vector $\vec{c} = C_1, C_2, ..., C_\Delta$
2. radius R

Under such definition, **all objects** $O_Y, O_Z, ...$ positioned **within volume of C are** to be understood as **members of C**.

END G-CATEGORY 3.2.2.0

We reinforce: under this view, a G_Δ-category is a convex region within S (Gärdenfors, 2004)[2]. Concrete geometric properties of such a ball (e.g. increase in volume in regards to increase of radius etc.) are, of course dependent on the nature of metric space in which the sphere is embedded (e.g. $V/r = 4/3\pi r^3$ for 3E-categories, i.e. categories embedded within 3-dimensional euclidean space).

In our simulations 2 and 3, we have used the Lightly Stochastic Binarization (Hromada, 2014b) algorithm to project initial objects onto positions within 128- or 64-dimensional binary Hamming spaces. We define categories within such spaces as follows:

H_Δ*-Category (DEF)*

H_Δ-Category is a Hamming ball within a Δ-dimensional Hamming space.

END H_Δ-CATEGORY 3.2.2.0

Given that

1. the radius of a H_Δ-Category cannot be higher than Δ (for such a sphere would envelop whole space S)
2. any integer Δ can be represented with $\log_2 \Delta$ bits
3. $\log_2 128 = 7$ and $\log_2 64 = 6$

it is evident that one needs exactly 135 bits of information[3] - in order to unambiguously specify a specific H_{128}-category embedded in a 128-dimensional hamming space.

And one needs 70 bits of information in order to unambiguously specify a H_{64}-category embedded in a 64-dimensional hamming space.

In this simulation, we shall juxtapose vectors representing diverse H_{64}-categories in order to obtain more complex schemata.

N_Δ*-Schema (DEF)*

An N_Δ-Schema is a result of concatenation of N vectors $\vec{g_1}, \vec{g_2}, \ldots, \vec{g_n}$ whereby each vector $\vec{g_1}, \vec{g_2}, \ldots, \vec{g_n}$ represents a G-category located within a Δ-dimensional space S_Δ.

END N_Δ-SCHEMA 3.2.2.0

Focus of the current simulation shall be on induction of schemata in case where $N = 4$. Given that basic units of such $4-$schemata will be H_{64}-categories, it can be easily seen that they such $4-$schemata could be encoded by no more and no less than $4 * 70 = 280$ bits.

END DEFINITIONS 3.2.2

[2] Those endowed synesthesia could potentially visualize G-categories as Δ-dimensional pearls (Hesse, 1967) or *balls of certain material, state and color.*

[3] 128 bits to specify coordinates of the centroid and 7 bits to specify the radius

Under these definitions, the model and the simulation described in this text can be understood as a method which aims to infer - from plain-text Corpus written in language L_{Corpus} - a 4—schema or (a set of 4—schemata) able to *generate* utterances which were originally not in the Corpus but are nonetheless still syntactically correct utterances of language L_{Corpus}. END INTRODUCTION 3.2

3.3 MODEL

In its essence, model presented in this simulation is reminiscent of the model presented in (Chapter 2). Hence, during the phase of "vector space preparation", texts from English-language transcripts of CHILDES corpora are first projected into 64—dimensional Hamming space H_{64} . Subsequently, a search within H_{64} is realized by means of an evolutionary algorithm.

There exists, however, a certain difference which ultimately causes the algorithm hereby presented to be essentially a non-supervised one. Thus, in the present situation, a H_X-category increase the probability of its survival in time if and only if is H_X contained in the utterance-like N—schema which matches as many utterances as possible.

3.3.1 VECTOR SPACE PREPARATION

Listing 7: PERL code of neighbor-word feature extractor

```
sub word_juxtaposition_featurefilter {
        my @features;
        my @all_words = split / /, shift;
        my $word = shift;
        my ($word_position)= grep { $all_words[$_] eq $word }
            0..$#all_words;
        if ($word_position==0) { #word begins the utterance
                push @features,"INIT";
                push @features,"1".$all_words[$word_position+1];

        } elsif ($word_position==$#all_words) { #word ends the
            utterance
                push @features,"-1".$all_words[$word_position-1];
                push @features,"END";
        } else {
                push @features,"-1".$all_words[$word_position-1];
                push @features,"1".$all_words[$word_position+1];

        }
        return @features;
}
```

Method known as Light Stochastic Binarization (LSB) (Hromada, 2014b) is used to project the input text onto H_{64}. Note, however, that initial features slightly differ from both approach presented in Chapter 1 which used word frequency distributions to project documents onto a resulting *semantic* space, as well as from approach presented in Chapter 2 which used suffixal information to project words onto a resulting *morphosemantic* space.

In contrast to both these methods, the feature extractor presented on Listing 7 focuses on two sources of information only: the identity of the word W_L and the word W_R juxtaposed to the left (resp. to the right) side of the target word W_X.

For example, the function call:

word_juxtaposition_featurefilter("this is a dog","dog")

returns array @features characterizing this concrete token of the word "dog" in terms of two features:

$$-1\mathtt{a},\ \mathsf{END}$$

In this case, the first feature encodes the fact that the token is preceded by an indeterminate article a while the second feature encodes the fact that "dog" is the last token of the utterance. Similarly, the token this would be characterized by features INIT, 1is; token is would be characterized by features $-1\mathtt{this}, 1\mathtt{a}$ and the token a would be characterized by features $-1\mathtt{is}, 1\mathtt{dog}$.

Once each word of each utterance is characterized by its features, one follows a standard Random Indexing procedure (Sahlgren, 2005) in order to attribute each distinct feature a distinct randomly generated $64-$dimensional sparsely non-zero "init" vector. Subsequently, euclidean representation of every word type W_X is obtained as a sum (i.e. unweighted linear combination) of features to which W_X is associated in the corpus.

These euclidean vectors are later normalized and enter the binarization procedure which leads to concise 8-byte hashes having the property:

> *The more words W_X and W_Y tend to occur in similar contexts, the less the Hamming distance between $\mathsf{LSB}(W_X)$ and $\mathsf{LSB}(W_Y)$ shall be.*

It is, indeed, this property which shall potentially allow us to effectuate successful evolutionary searches within the H_{64} space which could be potentially labeled as "morpho-syntactic".

END VECTOR SPACE PREPARATION 3.3.1

3.3.2 BRIDGING THE SUB-SYMBOLIC AND SYMBOLIC REALMS

In order to better understand the model hereby presented, one needs to understand a certain distinction often implemented by proponents of evolutionary programming (Fogel, 1995) or evolutionary strategies (Rechenberg, 1971). Id est, the distinction between the genotype and the phenotype.

Genotype

Information-encoding substrate potentially modifiable by variation and replication operators. Unambiguously translatable into phenotype.

END GENOTYPE 3.3.2.0

Phenotype

Concrete manifestation of specific genotype against which fitness can be evaluated. A distinct phenotype P_X can potentially manifest multiple distinct genotypes.

END GENOTYPE 3.3.2.0

Listing 8: Transcription of vector representations (genotypes) into regular expression phenotypes

```
$regex = "";
$extension = "";
for $component (0..5) {
        $component_regex = "";
        $component_extension = 0;
        $radius=$genotype_radius[$component];
        for $word (@all_words_in_corpus) {
                $word_hash=$word_hashes[$word]};
                $word_hcategory_distance = hamming_weight(
                    $word_hash XOR $genotype[$component]);
                if ($word_hcategory_distance<$radius) {
                        !$cregex ? ($cregex = ’(’.$word) : (
                            $cregex .= (’|’.$word));
                        $cextension++;
                }
        }
        $cregex ? ($regex .= ($cregex.’)’)):($regex .= ’’);
        $extension *= $cextension if ($cextension);
}
$regex=’^’.$regex.’$’; #utterance-based
```

In context of the current simulation, N−schemata (3.2.2) of length N = 4, i.e. 4−schemata, are to be understood as individual genotype instances. As is always the case in evolutionary computation,

Word	Hash	Word	Hash
this	BABA	that	BABB
it	BAAB	is	0F23
are	0F11	a	C123
the	C125	not	5FF5
duck	7720	dog	7725

Table 10: Words of a $\text{Corpus}_{\text{Mini}}$ and hexadecimal representations of their potential hashes.

Syntagma[5]	H_1		H_2		H_3		H_4		H_5	
	Center	Radius	Center	R	Center	R	Center	R	Center	R
	BABC	17	0F20	5	5FF0	7	C124	3	7723	7

Table 11: A candidate genotype which could be potentially induced from the hypothetic $\text{Corpus}_{\text{Mini}}$.

these schemata replicate, mutate, cross-over etc. But in order to get their fitness attributed, these genotypes have to be translated into phenotypes. Such translation is realized by means of the procedure displayed on Listing 8

The core idea of the genotype - phenotype translation is to be found on lines 9-11. On line 9, a hamming distance between hash of each among 5 components of the candidate genotype $4-schema$ is evaluated in regards to hash of each word W_X represented in the H_{64} vector space. On line 10, algorithm checks whether the obtained distance is smaller than the radius which is also included in the genotype. If yes, then the literal sequence of signifiant of the word W_X is injected into the resulting phenotype in a way, so that the resulting phenotype would be a syntactically correct Perl-Compatible Regular Expression (Wall et al., 1994; Hromada, 2011, 2016b) .

In other terms, the code displayed in Listing 8 can be understood as a method of translation of sub-symbolic (feature-based) binary vector representations into symbolic representations known as regular expressions.

For example, let's look at Table 10 which illustrates a small hypothetical $\text{Corpus}_{\text{Mini}}$ containing only words that, this, it, is ... and their corresponding binary hashes [4].

Then if ever a $5-schema$ like the one presented in Table 11 would be identified by the evolutionary search, it would be translated into a regular expression:

[4] As usual, 64-bit hashes are presented in hexadecimal format as sequences of four characters from range 0-9A-F

[5] In order to stay aligned with traditional linguistics, we shall sometimes use the term "syntagma" (resp. its abbreviated form "syn") as a synonym for the term "component".

^(this |that|it)(is)(not)(a |the)(dog |duck)$

which represents the microgrammar

$$\begin{gathered}\mathtt{Utterance} \rightarrow \mathtt{Syn_1 Syn_2 Syn_3 Syn_4 Syn_5} \\ \mathtt{Syn_1} \rightarrow \mathtt{this} \mid \mathtt{that} \| \mathtt{it} \\ \mathtt{Syn_2} \rightarrow \mathtt{is} \\ \mathtt{Syn_3} \rightarrow \mathtt{not} \\ \mathtt{Syn_4} \rightarrow \mathtt{a} \mid \mathtt{the} \\ \mathtt{Syn_5} \rightarrow \mathtt{dog} \mid \mathtt{duck}\end{gathered} \tag{6}$$

potentially covering 12 distinct utterances[6]. It would, however, not match utterances of a sort "this are not the dog" because the Hamming distance between the word $\mathtt{are}$ and the centroid of the 2nd component is bigger than the radius of the very same component (i.e. $HD(LSB("are"), Centroid_2) = HD(0F11, 0F20) = 9 > Radius_2$).

In such a way, one can determine the exact form of a Perl-Compatible regular expression (PCREs) by means of distance measurements in the underlying H_{64} space. And given that PCREs are

1. strings of symbols which describe sets of strings of symbols
2. a sort of *lingua franca* of many engineers active in the domain of Natural Language Processing, data-mining or information retrieval
3. well-tuned and optimized by almost three decades of development by not only PERL but also C++, Python, or R communities
4. transparent to inspections by human examinators[7]

one can potentially start to see a certain utility usefulness in developing an architecture which can unambiguously transform subsymbolic geometrized genotypes into comprehensible, symbolic, and manually modifiable PCRE-compatible phenotypes.

3.3.3 FITNESS FUNCTION

Fitness of N−schema N_X is principally determined by two characteristics:

[6] We shall further denote the quantity of *maximal theoretical number of covered utterances* with the term **extension**.

[7] Only 5 PCRE meta-characters are used in this article: **(** denotes beginning of a disjunctive group; **)** denotes end of a disjunctive group; | is a separator between two members of one a disjunctive group; ^ denotes beginning of expression and **$** denotes the end of expression

1. **extension** E, or a maximal theoretically possible sensitivity, is a finite natural number representing the quantity (i.e. the cardinality of a set) of all utterances which could be matched by N_X
2. Corpus **sensitivity** Y is a number of utterances, present in the Corpus, which have been matched by N_X

More formally: Let's have a $N-$schema X composed of N H_{64}-categories $H_{X1}, H_{X2}...H_{XN}$. Then X is said to have an overall extension E defined as a multiplicative product of extensions of individual categories:

$$E_X = \prod_{k=1}^{N} I_{H_k} \tag{7}$$

whereby the individual extension I_{H_k} of a $k-$th category H_k is defined as number of members of H_k. I.e. $|I_{H_k} = |H_k|$ where $|H_k|$ denotes the cardinality of set of objects whose distance from centroid $\vec{h_k}$ is less than the radius of category H_k.

For example, extension E of the $5-$schema presented in Table 11 is 12 because $I_{H_1} * I_{H_2} * I_{H_3} * I_{H_4} * I_{H_5} = 3 * 1 * 1 * 2 * 2 == 12$.

In contrast to E which is more an information-theoretic quantity, is the sensitivity Y a value which is always relevant in regards to certain corpus.

$$Y_X = N_X matchesCorpus$$

This notion is further exemplified by first line of following listing.

Extension and sensitivity thus defined, the fitness value of the schema N_X has been, for the purpose of the current simulation, defined as:

$$Fitness_1(N_X) = \frac{Y_X * Y_X}{E_X} \tag{8}$$

Rationale behind our choice of this and not other [8] is simple: given that we shall tend to maximize the fitness function, we put extension in the denominator (i.e. divisor) while putting the sensitivity into numerator (i.e. dividend).

Thus aligned, it may be expected that implementation of such a fitness function shall direct the evolutionary search towards schemata with both low extension as well as with high sensitivity. For this reason, sensitivity is squared in order to somewhat *counteract* the impact of extension which itself is a multiplicative product of its components.

[8] Many other fitness functions, of course, are possible and only very few of them have been tested. It cannot be excluded that more useful fitness functions shall be identified in the future. If not, then the fitness function $Fitness_1$ hereby defined could be potentially thought of as an expression of certain cognitive law. Such conjectures, however, would bring us too far.

3.3.4 EVOLUTIONARY STRATEGY

The INDUCTOR_1 evolutionary algorithm implemented in this simulation is similar to the algorithm CANONIC presented in subsec:optimization. Tournament operator is used as the main and only method of selection of fit individuals from the population to the mating pool. Size of the mating pool is equal to population size and mutations of centroid coordinates are equivalent to "bit flipping".

There exist, however, certain important differences which distinguish the algorithm hereby presented from the CANONIC:

1. implementation of phenotype-genotype distinction
2. evolution of both centroid coordinates as well as category radii
3. zeroth population is not generated pseudo-randomly
4. crossover occurs only at specific locations
5. re-focusing strategy is implemented

Taken together, this differences result in an algorithm endowed with certain characteristics of an evolutionary strategy (Rechenberg, 1971) or evolutionary programming (Fogel, 1995).

3.3.5 EVOLUTION OF BOTH CENTROIDS AND RADII

As had been already indicated, individual solutions identified by INDUCTOR_1 are essentially nothing else than $4-$schemata. That is, binary vectors which encode a syntagmatic sequence of four H_{64}-categories.

Given that a H_{64} category are defined in terms of both their center as well as radius, INDUCTOR_1 tries to identify not only the most optimal coordinates of category's centroid (as was the case in Chapter 2), but also the most optimal "extension" which is principally represented by H's radius.

Information about radius of each category is thus also part of the chromosome and is encoded as an integer value from range $< 0, \Delta >$. Probability of mutation of radius-encoding gene is 0.2%. If subjected to mutation, radius is either decremented or incremented with 1: this corresponds to category becoming less, resp. more exhaustive.

3.3.6 PSEUDO-RANDOM INITIALIZATION OF 0TH POPULATION

Every single individual of the initial population of $N-$schemata is generated as follows:

1. choose a random word W_1 occurring in the corpus and retrieve its geometric coordinates $\vec{w_1}$

2. define $\vec{w_R}1$ as the center of first category H_1
3. choose a random word W_2 occurring in the corpus and retrieve its geometric coordinates $\vec{w_2}$
4. define $\vec{w_2}$ as the center of the second component H_2
5. ...
6. choose a random word W_N occurring in the corpus and retrieve its geometric coordinates $\vec{w_N}$
7. define $\vec{w_N}$ as the center of the last syntagmatic component H_N

Subsequently, a radius which is neither too big nor too small is attributed to each among N components. In case of $\mathsf{INDUCTOR}_1$, the radius was set-up to value 13[9] which, in context to 64−dimensional Hamming space, seems to denote a distance which is neither too small nor too big.

Thus, contrary to *ex nihilo* initialization of CANONIC which started the induction process from randomly generated positions of all centroids, is $\mathsf{INDUCTOR}_1$'s initial 0-th population only partially random.

This is so because at the end of initialization process, center of each component of every individual N−schema is the same as the position of a certain word present in the $\mathtt{Corpus}$. [10]

3.3.7 LOCUS-CONSTRAINED CROSS-OVER

$\mathsf{INDUCTOR}_1$ cross-overs took place only at specific loci: namely at positions 64, 128 and 192 of the chromosome specifying centers of diverse $G - \mathtt{categories}$. In more practical terms, such a design choice assured that information precising all coordinates of $G - \mathtt{category}$ of the parent individual X have been substituted by information precising all coordinates of another $G - \mathtt{category}$ encoded in another parent individual Y.

This distinction aside, the usage of cross-over in $\mathsf{INDUCTOR}_1$ strategy has been fairly standard: every individual of a new generation was obtained as a result of cross-over between two randomly chosen members of the mating pool.

3.3.8 RE-FOCUSING STRATEGY

Another particular aspect is related to $\mathsf{INDUCTOR}_1$'s ability to prioritize, with every new run, induction of new schemata. In practice,

[9] Big radius results in big extension of the corresponding category and hence to many false positives. Small radius causes the category to have small extension and hence to potentially miss many true positives.

[10] Such an approach significantly boosts the inductive process which could have otherwise certain difficulties in *booting itself up*.

this is attained by starting every new run with execution of the code present in Listing 9.

Listing 9: PERL code behind re-focusing strategy

```
@corpus =grep {!/$previous_fittest_schema/ } @corpus;
```

Literally speaking, this line of code removes from the corpus all utterances matched by the most fittest $N - schema$ of the previous run. This results in gradual shrinking of size of the corpus against which the fitness of all future candidate schemata shall be evaluated.

In more general terms, the re-focusing strategy orients the process to *inference of schemata from such utterances, from which* ***no*** *schema has been yet induced.* [11].

And said in more "cognitive" terms, the algorithm invests more attention into exploration of structural regularities within data which have not yet been explored.

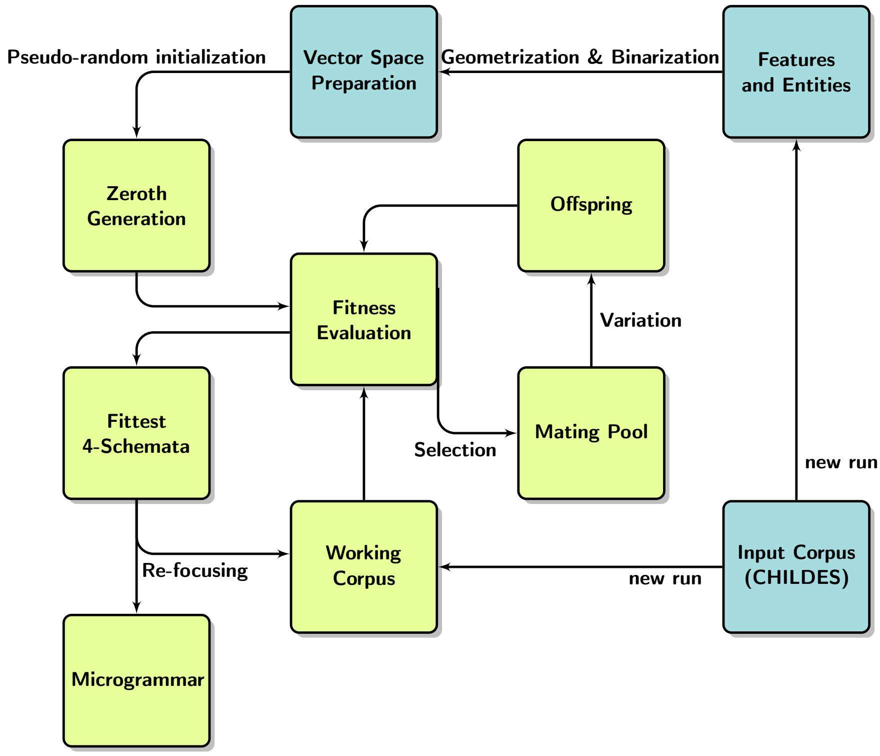

Figure 8: Data flow among main components of INDUCTOR. Lime color denotes components related to evolutionary optimization, aquamarine color denotes components of the preliminary VSP phase.

[11] Inductive process lacking the re-focus strategy would often "lock" itself to most salient patterns present in the corpus which would result in distinct runs often converging to similar schemata.

3.4 SIMULATION

Simulation presented in this section has implemented the evolutionary strategy INDUCTOR in order to induce sets of regexp-like rules from four-word English utterances contained in CHILDES corpus. Diagram elucidating relations between main INDUCTOR components is visible on Figure 8. The simulation was invoked twice, once in 64−dimensional space (INDUCTOR$_{64}$) and once in 128−dimensional space (INDUCTOR$_{128}$).

The vector space preparation phase (c.f. Section 3.3.1) yielded a vector space in which all subsequent INDUCTOR runs took place. Each among 2 * 100 distinct runs of INDUCTOR was initialized by a pseudo-random generation of zeroth population.

3.4.1 CORPUS

This article is conceived as a part of dissertation addressing the possibility of developing evolutionary models of induction of linguistic rules in (and by) human children. This makes the choice of the corpus quite straightforward: the corpus from which we shall aim to extract first linguistic categories is to be contained in Child Language Data Exchange System (CHILDES, (MacWhinney and Snow, 1985)).

Inspired by the "less is more hypothesis" (Elman, 1993), input corpus used in simulation hereby presented consisted of 1047 four-word "motherese"[12] utterances extracted from English section of CHILDES.

No other data has been used to guide the inductive process.

3.4.2 PARAMETERS

VSP	Input corpus	CHILDES$_{\text{English}}$[13]
	Feature Filter	word_juxtaposition
	Dimensionality	$\Delta = 64$ or $\Delta = 128$
	Seed	S = 3
	Reflections	I = 0
INDUCTOR	Population size	N = 100
	Selection	Tournament
	Crossover	One-point
	Mutation rate	M = 0.2%
	Initial population	pseudo-random
	Generations	G = 100
	Elitism	E = 0
	Runs	R = 100
Machine Learning	Syntagms	N = 4

Table 12: Parameters of diverse components of the INDUCTOR algorithm.

[12] In CHILDES, lines containing motherese utterances begin with the marker *MOT.

3.5 OBSERVATIONS

Appendix A lists 100 regexp-like rules which have been evaluated as "fittest" at the end of distinct INDUCTOR runs which took place in a H_{64} space. These hundred rules match 176 from 1047 utterances present in the input corpus (16.8%).

Appendix B lists 100 regexp-like rules which have been evaluated as "fittest" at the end of distinct INDUCTOR$_{128}$. These runs took place in a H_{128} space. These hundred rules match 176 from 1047 utterances present in the input corpus (15.8%).

As marked in both Appendices by the token GENERAL, INDUCTOR was also able to identify many completely grammatical 4-schemata which able to accept (or generate) even utterances which have not been present in the input corpus.

Such generalization faculty was observed in 82% resulting individuals in case of H_{64} and in 77% individual $4 - schemata$ induced in H_{128}.

Among these individuals induced in H_{64}, **32** have been manually evaluated as ALLGOOD, id est capable of accepting|generating only grammatically correct utterances of English language.

For example, the most fit schema of sixth run of INDUCTOR$_{64}$:

^(that)(is)(a)(bag|banana|basket|bridge|cherry|cow|gate |horse|kleenex|motorcycle|puzzle|rabbit|raccoon|shoe| spoon|story|timer|tractor)$

is able to accept|generate 18 grammatically correct English utterances in spite of the fact that only 5 among these 18 sentences have been explicitly present in the input corpus.

Excessive over-regularization was observed in case of 21 individuals willing to accept|generate at least one WRONG utterance.

Asides this, 4-schemata issued from 28 runs of INDUCTOR$_{64}$ have been marked as DISPUTABLE. That is, as capable of accepting|generating utterances which would be classified as "ungrammatical" by an orthodox grammarian, but could nonetheless occur in a real-life usage.

This border cases include utterances as:

where is the clever (individual 9)
what are we joey (individual 18)
there is what one (individual 34)
there does he go (individual 55)
what are you joey (individual 83)
oh what is i (individual 87)

as well as utterances which are syntactically correct, but semantically doubtful:

[13] Available at http://wizzion.com/thesis/simulation3/utterances.4

oh you are strawberries (individual 63)
oh you are fries (individual 63)
okay that is thumb (individual 91)

et caetera, et caetera.

In case of INDUCTOR$_{128}$ **37** induced 4—schemata have been manually evaluated as ALLGOOD and 17 as DISPUTABLE.

Listing 10: First exemplar of a non-monotonic ontogenetic trajectory

```
#ITERATION 30 FITNESS 1.333333
^(do )(you )(like )(candy|some|strawberries)$
#ITERATION 40 FITNESS 1.14285714285714
^(do )(you )(like )(bananas|box|candy|cover|fell|ketchup|nana|not
    |papa|popsicles|some|sorry|strawberries|tired)$
#ITERATION 50 FITNESS 1.8
^(do )(you )(like )(box|candy|ketchup|some|strawberries)$
```

3.5.1 DIACHRONIC OBSERVATIONS

A deeper time-oriented inspection of processes taking place during individual runs can also be of certain interest.

On Listing 10 it may be seen that after 30 iterations, INDUCTOR$_1$ has identified a 4-schema able to accept | generate utterances "do you like candy", "do you like some" and "do you like strawberries". However, this schema was lost in following 10 generations and fitness fell from 1.33 to 1.14[14]. Hence, an over-regular schema gained in prominence which was able to accept even such constructs as "do you like sorry" or "do you like tired".

But in following ten generations, population dynamics of the whole system not only lead to correction of the previous errors, but even brought about the increase in fitness to 1.8 which went hand in hand with scheme's ability to match utterances like "do you like box" or "do you like ketchup".

Another run presented on Listing 11 also exemplified such non-monotonic, error-correcting aspects of INDUCTOR$_1$ algorithm:

As it may be seen that an incorrect utterance "what is he going" was acceptable by the fittest individual of 40th and 50th iteration. This was corrected in 60th generation but further development brought about yet another batch of mistakes: utterances like "what is he cute" and "what is he share" were thus acceptable by the most fit individual of 80th generation. This has been subsequently corrected and the run terminated, after 100 generations, with a GENERAL, ALLGOOD 4-schema.

Listing 11: Second exemplar of a non-monotonic ontogenetic trajectory

[14] This is, of course, due to the fact that INDUCTOR$_1$ does not implement any form of elitism which would safeguard the fittest individuals from destructive variations.

```
#ITERATION 30 FITNESS 1.33333333333333
 ^(what )(is )(he )(doing|playing|saying)$
#ITERATION 40 FITNESS 1.8
 ^(what )(is )(he )(doing|going|holding|playing|saying)$
#ITERATION 50 FITNESS 1.5
 ^(what )(is )(he )(doing|drinking|going|holding|playing|saying)$
#ITERATION 60 FITNESS 1.8
 ^(what )(is )(he )(doing|drinking|holding|playing|saying)$
#ITERATION 70 FITNESS 2.25
 ^(what )(is )(he )(doing|holding|playing|saying)$
#ITERATION 80 FITNESS 2.28571428571429
 ^(what )(is )(he )(called|cute|doing|holding|playing|saying|
     share)$
#ITERATION 90 FITNESS 1.5
 ^(what )(is )(he )(doing|drinking|going|holding|playing|saying)$
#ITERATION 100 FITNESS 2.25
 ^(what )(is )(he )(doing|holding|playing|saying)$
```

3.6 CONCLUSION

Almost one third (32%) of $4 - schemata$ - identified by INDUCTOR$_1$ sweeping a 64—dimensional Hamming space representing 1047 English "motherese" utterances - produce only correct generalizations.

Collection of all induced N-schemata yields what we call a "microgrammar". Such a microgrammar is more a as construction-based (Fillmore et al., 1988; Lakoff, 1990) or usage-based (Tomasello, 2009) grammar than a grammar in sense of the Formal Language Theory (P117+122) or in the sense commonly accepted by proponents of the generativist doctrine (Chomsky, 2002).

But given that such a microgrammar (c.f. Appendix A) is capable of generating more syntactically correct utterances than those which had been presented through the training corpus, one can still consider it to be, in certain regards, modestly generative.

We say "modestly" because the generative faculty is kept on the leash by *evolution's tendency to discard such schemata which would be too concrete (i.e. have low sensitivity* Y*), or too exhaustive (i.e. have high extension* E*)*. Hence, the thorny problem of over-generalization is - at least in case of algorithm implementing the INDUCTOR$_1$ Evolutionary Strategy - not resolved by any *a priori* knowledge embedded in a some kind of chomskyan "Universal Grammar".

Far from it: we propose to depart from the idea that the grammar-inducing agents are not "ideal learners" in sense of Gold's Theorem (Gold, 1967; Johnson, 2004). On the contrary: the process of grammar-induction can only fully succeed if some information-encoding representations are, sometimes, irreversibly forgotten or subsumed to variation.

In this article, variation was attained by operators which:

1. mutate coordinates of centers of syntagmatic $G - categories$
2. mutate radii of syntagmatic $G - categories$ (i.e. increases or decreases category's extension)
3. substitute a $G - categories$ from one $N - schema$ with $G - categories$ from another $N - schema$ (i.e. locus-constrained crossover)

By causing these operators to perform their operations in a subsymbolic vector space, and by evaluating results of their activities on a symbol-sequence level, one can obtain a system able to induce simple $4 - schema$ microgrammars from simplified corpus of English "motherese" utterances which are four words long.

This[15], however, is only the beginning.

3.7 GENERAL DISCUSSION

> *There is an appealing symmetry in the notion that the mechanisms of natural learning may resemble the processes that created the species possessing those learning processes.*
>
> — D.E. Goldberg and J. Holland

More generally and beyond syntax, operators implemented in the 3rd simulation can be associated to following psychological phenomena:

1. mutation of an N-schema - synaptic pruning (P+38), information decay, forgetting etc.
2. crossover between two N-schemata - related to creativity, dreaming (P+89-90) and phantasia

Other variation operators - corresponding to certain forms of

- playing certain *language games* (Wittgenstein, 1953; Nowak et al., 1999), or "intrapsychic" (Brams, 2011) games
- imitating certain phenomena observed in linguistic behavior of human children (P+184-204)

could also be deployed.

Another subsequent enhancement of the GI method hereby introduced could potentially result from introduction of additional feature sets. For example, one could take a fit N−schema X, decompose it into its component G−categories $G_1, G_2, ..., G_N$ and, if ever a certain

[15] Proof-of-concept source code of this simulation is available at URL http://wizzion.com/thesis/simulation3/EGI.tgz under mrGPL license.

component G—category G_α turns out to be disjunctive, enrich vectorial representations of all its members with information that they belong to G_α. For example, one could enrich vectorial representations of tokens "doing", "holding", "playing", "saying" with information that they turned out to be subsumed under $G - category$ present in one quite fit $4 - schema$ (c.f. Listing 10). And enrich vectorial representations of tokens "ketchup", "strawberries" etc. with information that these tokens turned out to subsumed by yet another $G - category$ present in another schema (c.f. Listing 11).

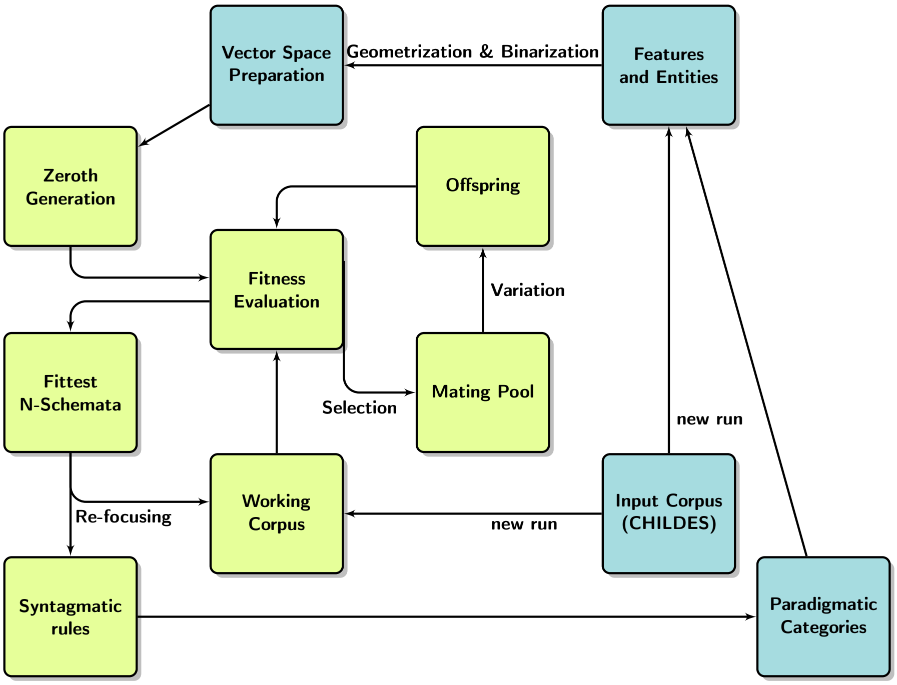

Figure 9: Data flow among main components of extended variant of INDUCTOR introducing a syntagmatic-paradigmatic feedback loop.

Note that introduction of such feature-sets could be interpreted as introduction of a feedback-loop in the system. Essence of such a system could thus be considered to be not only linguistic, but also cybernetic (Wiener, 1961; Lorenz, 1973). It could be postulated that introducing of such feed-back, bootstrapping (Hromada, 2014a; Karmiloff et al., 2009, pp.111-118) loop into the system would not only result in identification of more complex microgrammars, but would also cause the system to follow similar ontogenetic trajectories than those of children which undergo a so-called syntagmatic-paradigmatic shift (Nelson, 1977).

All such operators, features and feedback-loops taken together and coupled with

1. the fact that brain (P+5) is a finite material object with finite resources which is subjected to 2nd law of thermodynamics (P+7)

2. the fact that linguistic input which the child becomes is preprocessed by loving (P+241) and caring computational oracles (Turing, 1939; Clark, 2010) like mothers, fathers, care-takers etc.

3. the fact that acquisition of language takes place in informationally very rich, contextually grounded, usage-based scenarios (Tomasello, 2009)

one cannot exclude that a sort of evolutionary, ecological, equilibrium-seeking process indeed takes place in the mind of a modal healthy language-acquiring toddler.

And given that certain high-profile developmental linguists terminate their inquiry, concerning the informatic properties of the language input, with the conclusion

« internal mechanisms are necessary to account for the unlearning of ungrammatical utterances» (Marcus, 1993)

we allow us to conclude with a suggestion that the *internal mechanism* which Marcus mentions is, in reality, not a sort of universal grammar (P+98-101) black-box but instead a potentially "general cognitive process" (P+101, (Piaget, 1974)) whose very essence is to discard that, which is non-functional:

Evolution (P+3).

3.8 THIRD SIMULATION BIBLIOGRAPHY

Brams, S. J. (2011). *Game theory and the humanities: bridging two worlds*. MIT Press.

Brodsky, P., Waterfall, H., and Edelman, S. (2007). Characterizing motherese: On the computational structure of child-directed language. In *Proceedings of the 29th Cognitive Science Society Conference, ed. DS McNamara & JG Trafton*, pages 833–38.

Chomsky, N. (2002). *Syntactic structures*. Walter de Gruyter.

Clark, A. (2010). Distributional learning of some context-free languages with a minimally adequate teacher. In *Grammatical Inference: Theoretical Results and Applications*, pages 24–37. Springer.

Fillmore, C. J., Kay, P., and O'connor, M. C. (1988). Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, pages 501–538.

Fogel, D. B. (1995). Phenotypes, genotypes, and operators in evolutionary computation. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 1, page 193. IEEE.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). Artificial intelligence through simulated evolution.

Gold, E. M. (1967). Language identification in the limit. *Information and control*, 10(5):447–474.

Goldberg, D. E. and Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3:95–99.

Hesse, H. (1967). *Das Glasperlenspiel: Versuch e. Lebensbeschreibung d. Magisters Ludi Josef Knecht samt Knechts hinterlassenen Schriften*, volume 842. Suhrkamp.

Hromada, D. D. (2011). Initial experiments with multilingual extraction of rhetoric figures by means of perl-compatible regular expressions. In *RANLP Student Research Workshop*, pages 85–90.

Hromada, D. D. (2014a). Conditions for cognitive plausibility of computational models of category induction. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 93–105. Springer.

Hromada, D. D. (2014b). Empiric introduction to light stochastic binarization. In *Text, Speech and Dialogue*, pages 37–45. Springer.

Hromada, D. D. (2016a). Evolutionary induction of 4-schema micro-grammars from childes corpora. submitted to journal Evolutionary Computation.

Hromada, D. D. (2016b). Reproducible identification of pragmatic universalia in childes transcripts. Accepted for JADT2016 conference.

Johnson, K. (2004). Gold's theorem and cognitive science. *Philosophy of Science*, 71(4):571–592.

Karmiloff, K., Karmiloff-Smith, A., and Karmiloff, K. (2009). *Pathways to language: From fetus to adolescent*. Harvard University Press.

Lakoff, G. (1990). *Women, fire, and dangerous things: What categories reveal about the mind*. Cambridge Univ Press.

Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition*, 46(1):53–85.

Nelson, K. (1977). The syntagmatic-paradigmatic shift revisited: a review of research and theory. *Psychological bulletin*, 84(1):93.

Nowak, M. A., Plotkin, J. B., and Krakauer, D. C. (1999). The evolutionary language game. *Journal of Theoretical Biology*, 200(2):147–162.

Rechenberg, I. (1971). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dr.-Ing*. PhD thesis, Thesis, Technical University of Berlin, Department of Process Engineering.

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5.

Solan, Z., Horn, D., Ruppin, E., and Edelman, S. (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11629–11634.

Tomasello, M. (2009). *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press.

Turing, A. M. (1939). Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 2(1):161–228.

Wall, L. et al. (1994). The perl programming language.

Wiener, N. (1961). *Cybernetics or Control and Communication in the Animal and the Machine*, volume 25. MIT press.

Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell.

Wolff, J. G. (1988). Learning syntax and meanings through optimization and distributional analysis. *Categories and processes in language acquisition*, 1(1).

4 SUMMA

> *The natural selection paradigm of such knowledge increments can be generalized to other epistemic activities, such as learning, thought and science.*
>
> — D.T. Campbell

The objective of this dissertation was to provide a computational evidence of the "operational thesis" (P+20):

«Learning of toddlerese can be successfully simulated by means of evolutionary algorithm processing textual representations of motherese.»

Given that

- the third simulation used no other input than the plain-text corpus of motherese utterances

and given that

- the third simulation resulted in identification of schemata able to generate grammatically correct utterances which have not been present in the initial corpus

one may consider the "operational thesis" as temporarily unfalsified.

A Popperian conclusion

In this sense, we consider any future effort to falsify or verify "the softest thesis" (P+17-19):

«Ontogeny of toddlerese can be successfully simulated by means of evolutionary computation.»

as effort worthy of interest.

It is worthy of noting in this regards that certain notions like that of a $4-schema$ or *morphosemantic class* are not to be considered as some ultimate elements of some sort of *ewige Theorie* but rather as temporary, limited building blocks of an architecture which is to be surpassed.

Surpassed by what? Maybe surpassed by models which introduce not only $4-schemata$ but also $2-schemata$, $3-schemata$, $5-schemata$ ... $N-schemata$. Or by procedures which integrate semantic, morphological and syntactic spaces within a single "linguistic" space S_L. Given what we have seen until now, it can not be a priori excluded that results of certain types of evolution-inspired simulations taking place within such S_L would turn out to be consistent with "the softer hypothesis" (P+14-16) which states that

«learning of natural language can be successfully simulated
by means of evolutionary computation»

But when speaking about optimization taking place within a linguistic space S_L, shouldn't it be also possible to speak also about optimization taking place within even more generic a space S_G ? For nothing prohibits that category-inducing methods hereby introduced could be used to induce classifiers of partially or even fully non-linguistic entities. For example, a research project stemming from this dissertation may potentially explore the extent in which the evolutionary search for prototypes could be useful in Computer Vision: the only thing which would be fundamentally different would be the essence of input entities (i.e. images and not texts) and features occurring in such entities (e.g. Haar features (Viola and Jones, 2001; Hromada, 2010; Hromada et al., 2010) or others).

Relation to Computer Vision

In fact, nothing forbids to use one among three CI models hereby introduced whenever one needs to perform:

1. multiclass classification of entities (exemplified by "supervised" simulations 1 and 2)
2. induction of rules from positive corpus only (exemplified by "unsupervised" simulation 3)

In other terms, the combination of "vector spaces" and "evolutionary computation" components can be understood as a "generic optimization toolbox" (GOT) which could potentially be applied upon any set of features. It is, however, primarily the nature of the input corpus and the nature of features which extracted from the corpus which should most closely determine the nature of categorization-performing agent thus induced.

Generic Optimization Toolbox

Hence, when applied upon data-sets describing "spatial" trajectories within a group of "labyrinths", one could aspire to induce rules allowing a certain robot, a certain automatized vehicle, or a certain sort of embedded artificial classifier system (Booker et al., 1989), to find its way out of the "labyrinth" it never saw before.

Induction of spatial trajectories

Or - if one would depart from so-called "morally relevant features" (Hromada and Gaudiello, 2014) - one could even hope to simulate ontogeny of categories and rules of a somewhat different kind. That is, of categories and rules which are commonly labeled as "aesthetic" (i.e. beautiful / ugly), "moral" (i.e. good / bad), "deontologic" (i.e. forbidden / allowed) (Hromada, 2016).

Moral Induction

Asides "linguistic", "visual", "spatial" or "moral", implementation of EML GOTs in induction of other types of intelligence (Gardner, 2011) or their combinations (Karpathy and Fei-Fei, 2015) in artificial agents and robots is also a task to be explored. If successful, it cannot be excluded that such explorations would potentially bring scientific and engineering communities one step closer us to deployment meta-modular (Hromada, 2012) artificial agents able to:

EML and theory of multiple intelligences

1. integrate (Tononi, 2004) multi-modal (i.e. linguistic, visual, proprioceptive etc.) information
2. use nature-inspired, evolutionary computational core to identify most fit groupings of such information

By doing so, an ultimate *ex computatio atque simulatio* proof of the "soft thesis" (P+11-13):

> «learning can be successfully simulated
> by means of evolutionary computation »

could be, potentially, given.

To offer such a proof, however, is a task which by far surpasses limits of any individual researcher. What is more, alternative machine learning paradigms (e.g. deep learning) currently predominate and it may be the case that popularity of such approaches decreases the amount of attention which could - and should - be focused on exploration of common grounds between computational models of learning and computational models of evolution.

Let's now enumerate certain advantageous properties of evolutionary machine learning (EML) models which have been presented in simulations one, two and three. These EML models are :

Evolutionary Machine Learning and its advantages

1. **functional**: function of the model is principally determined by choice of fitness function and selection/variation operators
2. **alternative**: in any moment T_X, the learning system contains multiple alternative solutions of the problem (P+8-10)
3. **population-based**: behavior of the learning system can be interpreted in terms of population dynamics (P+116)

Contrary to these, connectionist models are more "structural" than "functional", they do not explicitly encode representations of diverse solutions and their convergence towards optimal states is more easily interpretable in terms of differential "gradient descent" of "backpropagation" than in terms of population dynamics.

Comparison with connectionist models

What's more, by coupling the notion of evolution with that of a vector space, and by implementing a fairly trivial phenotype - genotype transcription (Section 3.3.2), one can obtain unsupervised EML models

1. bridging sub-symbolic (vectorial) and symbolic (regexps and grammars) realms
2. transparent to investigation and modulation by a human investigator (i.e. easy to interpret and *teach*)

Note that the property of being transparent to investigation and modulation is not a property which should be taken *à la légère*. For it could result in a creation of the inter-subjective bound between the artificial system which is being (investig | modul)ated and the human who (investig | modul)ates.

Of learners and teachers

In other terms, it could, potentially, result in emergence of entities of non-organic origin *who* could, and should, be considered as not only *objects of machine-learning* but also as *subjects of machine-teaching*.

Such considerations, however, bring us further than paradigms like machine learning or even computer science could ever bring us. Such considerations bring us towards meta-paradigm[1] of paedagogy and didactics (Komenský et al., 1991) which solely can demonstrate the validity and usefulness of the Theory of Intramental Evolution (Hromada, 2015).

Such considerations bring us towards such regions of S_G whereby the very "hard thesis" (P+2-10)

«Learning is a form of evolution»

could be evaluated as valid.

Valid or not, nothing forbids the sign-manipulating[2] mind (P+1) to realize a transposition (P+190-192) which savants like Bateson (Bateson, 2006) once realized.

Ultimate Chiasm

That is, a transposition between two terms each of which denote one big stochastic system, a transposition between "Mind" and "Nature", a transposition which obliges one to state:

«Evolution is a form of learning[3]»

Such is, indeed, the ultimate result of the dissertation with which we aspire for attribution of the title *Philosophiae Doctor* in both cybernetics as well as cognitive psychology.

Such is, indeed, the result of work commenced by two words forming the "initial thesis" (P+1):

«Mind Evolves»

*
* *

[1] A scientific paradigm (Kuhn, 2012) transfers knowledge about certain field of study. A scientific meta-paradigm transfers knowledge concerning the transfer of knowledge.

[2] « Thinking is essentially the activity of operating with signs.» (Wittgenstein, 1934)

[3] Lorenz (1973) states that the principal difference between learning and evolution is the ability of a learning system to "learn from one's own errors". System which learns is supposed to have such ability while system which "only" evolves does not. But is it really always the case?

4.1 summa bibliography

Bateson, G. (2006). Mind and nature: A necessary unity (advances in systems theory, complexity, and the human sciences).

Booker, L. B., Goldberg, D. E., and Holland, J. H. (1989). Classifier systems and genetic algorithms. *Artificial intelligence*, 40(1):235–282.

Campbell, D. T. (1974). An essay on evolutionary epistemology. *The philosophy of Karl Popper*, pages 413–463.

Gardner, H. (2011). *Frames of mind: The theory of multiple intelligences*. Basic books.

Hromada, D. D. (2010). smiled : Sourire naturel et sourire artificiel. de l'utilisation d'opencv pour le tracking, la reconnaissaince des expressions faciales et la détection du sourire. Master's thesis, Ecole Pratique des Hautes Etudes, Paris, France.

Hromada, D. D. (2012). From age&gender-based taxonomy of turing test scenarios towards attribution of legal status to meta-modular artificial autonomous agents. In *Revisiting Turing and His Test: Comprehensiveness, Qualia and the Real World*, pages 7–11. AISB and IACAP Turing Centennary World Congress, Birmingham, United Kingdom.

Hromada, D. D. (2015). *Conceptual Foundations : Intramental Evolution & Ontogeny of Toddlerese*. Propedeutica Didactica. in print. Supplementary material for PhD. dissertation.

Hromada, D. D. (2016). Narrative fostering of morality in artificial agents: Constructivism, machine learning and story-telling. In *L'esprit au-delà du droit: Pour un dialogue entre les sciences cognitives et le droit*. Mare et Martin.

Hromada, D. D. and Gaudiello, I. (2014). Introduction to moral induction model and its deployment in artificial agents. In *Sociable Robots and the Future of Social Relations*, pages 209–216. IOS Press.

Hromada, D. D., Tijus, C., Poitrenaud, S., and Nadel, J. (2010). Zygomatic smile detection: The semi-supervised haar training of a fast and frugal system: A gift to opencv community. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on*, pages 241–245. IEEE.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.

Komenskỳ, J. A., Okál, M., and Pšenák, J. (1991). *Vel'ká didaktika: Didactica magna*. Slovenské pedagogické nakladatel'stvo.

Kuhn, T. S. (2012). *The structure of scientific revolutions*. University of Chicago press.

Lorenz, K. (1973). *Die Rückseite des Spiegels*. R. Piper.

Piaget, J. (1974). Introduction à l'épistémologie génétique. *Paris, PUF*.

Tononi, G. (2004). An information integration theory of consciousness. *BMC neuroscience*, 5(1):42.

Wittgenstein, L. (1934). The blue book.

A

APPENDIX: LIST OF 4–SCHEMAS INDUCED IN H_{64}

Following list enumerates one hundred individual 4-schemas issued from one hundred distinct runs of INDUCTOR$_1$ Evolutionary Strategy. Induction took place in 64–dimensional Hamming space.

The token GENERAL denotes a 4-schema (accep|genera)ting more utterances than were present in the training corpus (i.e. extension E is bigger than sensitivity Y).

The token ALLGOOD is attributed to such GENERAL 4-schemas which accept|generate only correct English sentences.

Listing 12: Hundred 4-schemas induced in 64-dimensional binary space

```
^(that )(is )(a )(bird|cat|fence|piece|radio|teapot|tractor)$ #E
    =7,Y=6,GENERAL,ALLGOOD
^(it )(is )(a )(bag|banana|basket|belly|bird|boat|bottle|bridge|
    brush|butterfly|calf|carnival|cherry|comb|cone|cow|days|doc|
    dwarfs|face|fish|goat|grandmother|horse|hose|kleenex|mess|
    motorcycle|mountain|mushroom|pear|pen|picture|pig|rabbit|
    refill|robot|rock|sailboat|second|sheep|squirrel|story|teddy|
    telephone|timer|tomato|tractor|train|truck|worm)$ #E=51,Y=16,
    GENERAL,3WRONG
^(where )(is )(the )(baby|bag|comb|fork|grandma|phone|raccoon|
    spoon|surprise|teapot|teaspoon|tractor)$ #E=12,Y=5,GENERAL,
    ALLGOOD
^(what )(are )(you )(calling|doing|drawing|holding|making|saying|
    throw)$ #E=7,Y=4,GENERAL,1WRONG
^(that )(is )(a )(boy|brush)$ #E=2,Y=2
^(that )(is )(a )(bag|banana|basket|bridge|cherry|cow|gate|horse|
    kleenex|motorcycle|puzzle|rabbit|raccoon|shoe|spoon|story|
    timer|tractor)$ #E=18,Y=5,GENERAL,ALLGOOD
^(what )(is )(that )(cho|for|neat|piece)$ #E=4,Y=2,GENERAL,1WRONG
^(is )(that )(a )(car|hat)$ #E=2,Y=2
^(where )(is )(the )(airplane|clever|farm)$ #E=3,Y=2,GENERAL,1
    DISPUTABLE
^(what )(is )(on |spoon )(there)$ #E=2,Y=1,GENERAL,1WRONG
^(hang |yeah )(and )(this )(one)$ #E=2,Y=1,GENERAL,1WRONG
^(this )(is )(a )(calf|cherry)$ #E=2,Y=2
^(where )(is )(the )(bath|blocks|bucket|cover|rest|rocks|socks|
    teaspoon|water)$ #E=9,Y=5,GENERAL,2WRONG
^(you )(can |help )(do )(it)$ #E=2,Y=1,GENERAL,ALLGOOD
^(is )(that )(a )(comb|cone)$ #E=2,Y=2
^(you )(gonna )(do |try )(it)$ #E=2,Y=1,GENERAL,ALLGOOD
^(there )(is )(a )(car|mountain|train)$ #E=3,Y=3
^(what )(are )(we )(drinking|joey|playing)$ #E=3,Y=2,GENERAL,1
    DISPUTABLE
^(there |where )(are )(you )(going)$ #E=2,Y=1,GENERAL,ALLGOOD
```

```
^(oh )(are )(you )(feet|okay)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(it )(is )(a )(baby|puzzle)$ #E=2,Y=2
^(where )(is )(the )(hat|truck)$ #E=2,Y=2
^(or |she )(is )(the )(baby)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(oh )(here |there )(it )(is)$ #E=2,Y=2
^(what )(is )(this )(called|miss|puzzle)$ #E=3,Y=2,GENERAL,
    ALLGOOD
^(what )(is )(he )(holding|saying)$ #E=2,Y=2
^(there |where )(do )(we )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(is )(nt )(that )(cool|great)$ #E=2,Y=2
^(and )(what )(is )(book|that)$ #E=2,Y=1,GENERAL,ALLGOOD
^(where )(do |try )(they )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(what )(do )(they )(do|out)$ #E=2,Y=1,GENERAL,1WRONG
^(there |where )(is )(it )(going)$ #E=2,Y=1,GENERAL,ALLGOOD
^(oh )(here |there )(you )(go)$ #E=2,Y=2
^(there |where )(is )(what )(one)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(honey |yeah )(that )(is )(it)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(is )(about |that )(there)$ #E=2,Y=1,GENERAL,1WRONG
^(what )(do )(you )(say|take)$ #E=2,Y=1,GENERAL,ALLGOOD
^(it )(is )(nosed |red )(too)$ #E=2,Y=1,GENERAL,ALLGOOD
^(where )(does |goes )(that )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(where )(is )(your )(cup|pants)$ #E=2,Y=1,GENERAL,1WRONG
^(yeah )(it )(is )(feet|okay)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(that )(is )(not )(juice|nice|toast)$ #E=3,Y=3
^(where )(is )(the )(boy|man|milk)$ #E=3,Y=2,GENERAL,ALLGOOD
^(and )(some |there )(you )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(this )(is )(baby |for )(is)$ #E=2,Y=1,GENERAL,1WRONG
^(what )(is )(it )(grandmaman|maman)$ #E=2,Y=0,GENERAL,ALLGOOD
^(that )(is )(a )(hat|truck)$ #E=2,Y=1,GENERAL,ALLGOOD
^(okay )(that )(does |want )(it)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(okay )(some |there )(you )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(does |want )(it )(go )(here)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(there )(is )(a )(fish|horsie|raccoon|spoon)$ #E=4,Y=3,GENERAL
^(what )(do )(you )(think|want)$ #E=2,Y=2
^(okay )(here |there )(we )(go)$ #E=2,Y=2
^(and )(look |oh )(and )(this)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(there |where )(does )(he )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(that )(is )(baby |for )(is)$ #E=2,Y=1,GENERAL,1WRONG
^(it )(is )(missing |not )(there)$ #E=2,Y=1,GENERAL,ALLGOOD
^(and )(this )(in |juice )(here)$ #E=2,Y=1,GENERAL,ALLGOOD
^(do )(you )(like )(bananas|strawberries|thumb|tired)$ #E=4,Y=2,
    GENERAL,1WRONG
^(are )(you )(my )(daddy|tea)$ #E=2,Y=0,GENERAL,ALLGOOD
^(where )(does |want )(this )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(and )(what )(are )(destructo|these)$ #E=2,Y=1,GENERAL,1
    DISPUTABLE
^(oh )(you )(are )(busy|distracted|finished|fries|funny|
    strawberries|tara)$ #E=7,Y=2,GENERAL,2DISPUTABLE
^(what )(is )(he )(doing|saying)$ #E=2,Y=2
^(you )(are )(all )(done|gone|sticky)$ #E=3,Y=2,GENERAL,ALLGOOD
^(what )(does )(he )(get|say|take)$ #E=3,Y=3
^(where )(is )(going |he )(going)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(there )(is )(a )(face|piece)$ #E=2,Y=2
```

```
^(what )(is )(baby )(doing|pieces)$ #E=2,Y=1,GENERAL,1WRONG
^(do )(you )(got |upside )(it)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(there |where )(are )(they )(going)$ #E=2,Y=1,GENERAL,1
    DISPUTABLE
^(can |love )(you )(do )(it)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(you )(are )(so )(mean|nice)$ #E=2,Y=1,GENERAL,ALLGOOD
^(she )(is )(almost |must )(there)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(is )(it )(for |on )(me)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(is )(this )(here|there)$ #E=2,Y=1,GENERAL,ALLGOOD
^(where )(does |want )(it )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(do )(nt )(dump |kick |throw )(it)$ #E=3,Y=2,GENERAL,ALLGOOD
^(ah |yeah )(you )(see )(them)$ #E=2,Y=1,GENERAL,ALLGOOD
^(that )(is )(the )(blocks|daddy|sugar)$ #E=3,Y=2,GENERAL,ALLGOOD
^(it )(is )(a )(rabbit|raccoon)$ #E=2,Y=2
^(is )(that )(another |some )(baby)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(are )(you )(joey|playing)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(what )(is )(the )(air|around|bath|bike|copter|dog|puzzles|rest)
    $ #E=8,Y=2,GENERAL,2WRONG
^(okay |some )(you )(brush )(it)$ #E=2,Y=1,GENERAL,1WRONG
^(some |where )(are )(we )(going)$ #E=2,Y=1,GENERAL,1WRONG
^(oh )(what )(is )(i|that)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(it )(is )(for |go )(me)$ #E=2,Y=1,GENERAL,1WRONG
^(she )(is )(all )(naked|sticky)$ #E=2,Y=1,GENERAL,ALLGOOD
^(that )(is )(your )(cup|pants)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(okay )(that )(is )(delicious|enough|good|thumb)$ #E=4,Y=2,
    GENERAL,ALLGOOD
^(you )(finishes |kwe |want )(this )(one)$ #E=3,Y=1,GENERAL,2
    WRONG
^(is )(that )(a )(fireman|man)$ #E=2,Y=1,GENERAL,ALLGOOD
^(and )(what )(is )(stuck|this)$ #E=2,Y=1,GENERAL,ALLGOOD
^(try )(it )(on |spoon )(there)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(where )(is )(your )(good|xxx)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(did |uhoh )(she )(do)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(it )(is )(mickey |right )(there)$ #E=2,Y=1,GENERAL,ALLGOOD
^(feed )(the )(baby )(feet|okay)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(what )(is )(that )(great|noise)$ #E=2,Y=1,GENERAL,1DISPUTABLE
```

B

APPENDIX: LIST OF 4−SCHEMAS INDUCED IN H_{128}

Following list enumerates one hundred individual 4-schemas issued from one hundred distinct runs of INDUCTOR$_1$ Evolutionary Strategy. Induction took place in 128−dimensional Hamming space.

Listing 13: Hundred 4-schemas induced in 128-dimensional binary space

```
^(where )(is )(the )(bird|farm|teapot|tractor)$ #E=4,Y=4
^(it )(is )(a )(any|bad|basket|bat|bed|boat|bottle|brush|
    butterfly|camera|carnival|comb|cone|cow|dress|elmo|enough|
    fence|glass|goat|grandmother|hips|horse|kay|lamp|mess|mine|
    motorcycle|mountain|nail|pants|pear|pen|picture|ponytail|
    rabbit|raccoon|sailboat|sentences|sheep|sophie|squirrel|
    stomach|swim|teacup|teddy|telephone|timer|tired|tomato|
    trailer|train|tunnel|turn|turtle|working|worm)$ #E=57,Y=15,
    GENERAL,4WRONG
^(oh )(what )(is )(that|this)$ #E=2,Y=2
^(that )(is )(a )(alone|bag|be|bird|box|car|carnival|cone|farm|
    fork|gate|house|jello|jure|kleenex|motorcycle|neighbor|others
    |phone|plates|purple|puzzle|rabbit|raccoon|radio|road|someone
    |stairs|story|teaspoon|telephone|tractor|world)$ #E=33,Y=8,
    GENERAL,2WRONG
^(what )(is )(he )(doing|saying)$ #E=2,Y=2
^(okay )(here |there )(we )(go)$ #E=2,Y=2
^(what )(do )(you )(say|want)$ #E=2,Y=2
^(what )(does )(he )(say|take)$ #E=2,Y=2
^(do )(this )(anyone |one )(now)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(what )(do )(they )(do|mean)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(are )(you )(calling|doing|drawing|making|saying)$ #E=5,Y
    =4,GENERAL,ALLGOOD
^(do )(you )(like )(frosting|some|strawberries)$ #E=3,Y=3
^(almost |where )(is )(your )(xxx)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(what )(is )(she )(carrying|drinking|winking)$ #E=3,Y=2,GENERAL,
    ALLGOOD
^(where )(is )(the )(airplane|blocks|box|car|cover|daddy|eyes|
    grandmother|hat|idea|neighbor|plates|purple|rest|road|socks|
    stairs|teaspoon|together|truck)$ #E=20,Y=7,GENERAL,1WRONG
^(what )(are )(we )(drinking|playing)$ #E=2,Y=2
^(where )(is )(the )(baby|man)$ #E=2,Y=2
^(that )(is )(baby |really )(is)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(gonna |where )(does )(this )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(i )(fell |will )(try )(xxx)$ #E=2,Y=0,GENERAL,1WRONG
^(where )(is )(the )(milk|water)$ #E=2,Y=2
^(gonna |where )(does )(what )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(there )(is )(a )(hips|mountain|train)$ #E=3,Y=2,GENERAL,1WRONG
^(oh )(here |there )(it )(is)$ #E=2,Y=2
```

```
^(is )(that )(a )(bird|car|hat|man)$ #E=4,Y=4
^(may |yeah )(that )(is )(it)$ #E=2,Y=1,GENERAL,1WRONG
^(how )(are )(you )(doing|saying)$ #E=2,Y=1,GENERAL,ALLGOOD
^(ah )(you )(are )(carrying|drinking|winking)$ #E=3,Y=1,GENERAL,
    ALLGOOD
^(is )(it )(for )(her|me)$ #E=2,Y=2
^(can |have )(i )(try )(it)$ #E=2,Y=0,GENERAL,1DISPUTABLE
^(is )(it )(closed |cold )(now)$ #E=2,Y=2
^(really |so )(is )(this )(you)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(is )(this )(called|puzzle)$ #E=2,Y=2
^(okay )(that )(does |want )(it)$ #E=2,Y=1,GENERAL,1WRONG
^(what )(is )(baby )(doing|going)$ #E=2,Y=1,GENERAL,1WRONG
^(where )(is )(he )(going|playing)$ #E=2,Y=1,GENERAL,ALLGOOD
^(it )(is )(for )(her|me)$ #E=2,Y=1,GENERAL,ALLGOOD
^(okay )(like |upside )(that )(here)$ #E=2,Y=1,GENERAL,1
    DISPUTABLE
^(gonna |where )(does )(he )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(it )(is )(for |on )(you)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(can )(we )(make|take)$ #E=2,Y=1,GENERAL,ALLGOOD
^(almost |where )(do )(we )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(you )(gonna |where )(do )(it)$ #E=2,Y=1,GENERAL,1WRONG
^(and )(who )(is )(strawberries|that)$ #E=2,Y=1,GENERAL,1
    DISPUTABLE
^(that )(is )(a )(boy|brush|fence|kleenex|motorcycle|puzzle|spot|
    teapot)$ #E=8,Y=6,GENERAL,ALLGOOD
^(what )(cold |does )(it )(do)$ #E=2,Y=1,GENERAL,1WRONG
^(i )(open |put )(it )(down)$ #E=2,Y=1,GENERAL,ALLGOOD
^(and )(what )(is )(that|this)$ #E=2,Y=2
^(i )(put )(honey |mickey )(xxx)$ #E=2,Y=0,GENERAL,ALLGOOD
^(really |so )(who )(are )(you)$ #E=2,Y=1,GENERAL,ALLGOOD
^(that )(is )(the )(daddy|sugar|water)$ #E=3,Y=2,GENERAL,ALLGOOD
^(this )(anyone |one )(goes )(there)$ #E=2,Y=1,GENERAL,1
    DISPUTABLE
^(almost |how )(does )(this )(go)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(oh )(here |there )(you )(go)$ #E=2,Y=2
^(this )(is )(baby |really )(is)$ #E=2,Y=1,GENERAL,1WRONG
^(okay )(there |yes )(you )(go)$ #E=2,Y=1,GENERAL,ALLGOOD
^(that )(is )(his )(shoe|sister|tail)$ #E=3,Y=3
^(bring )(it )(back |pick )(here)$ #E=2,Y=1,GENERAL,1WRONG
^(can )(i )(have |love )(it)$ #E=2,Y=0,GENERAL,ALLGOOD
^(what )(is )(that )(cho|for|noise)$ #E=3,Y=2,GENERAL,ALLGOOD
^(do )(nt )(dump |throw )(it)$ #E=2,Y=2
^(do )(you )(like )(bananas|firemen|that|us)$ #E=4,Y=2,GENERAL,
    ALLGOOD
^(where )(are )(you )(going|playing)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(can |should )(we )(do)$ #E=2,Y=2
^(i )(fell |will )(show )(you)$ #E=2,Y=0,GENERAL,1WRONG
^(you )(go )(like |those )(this)$ #E=2,Y=1,GENERAL,1WRONG
^(gonna |where )(do )(they )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(you )(are )(almost |not )(here)$ #E=2,Y=1,GENERAL,ALLGOOD
^(you )(can )(do )(that|this)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(do )(we )(say|take)$ #E=2,Y=1,GENERAL,ALLGOOD
^(she )(is )(almost |really )(there)$ #E=2,Y=1,GENERAL,ALLGOOD
```

```
^(okay )(you )(brush |bunny )(it)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(where )(is )(it )(going|playing)$ #E=2,Y=1,GENERAL,ALLGOOD
^(yeah )(it )(is )(okay|turn)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(xxx )(this )(anyone |one )(here)$ #E=2,Y=1,GENERAL,1WRONG
^(for )(the )(baby |really )(okay)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(oh )(are )(you )(hair|okay)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(where )(are )(they )(going|playing)$ #E=2,Y=1,GENERAL,ALLGOOD
^(look )(here |there )(is )(xxx)$ #E=2,Y=1,GENERAL,ALLGOOD
^(there )(is )(a )(head|hips|horsie|name|picture|trailer)$ #E=6,Y
    =2,GENERAL,ALLGOOD
^(there |yes )(is )(your )(tea)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(what )(we )(gonna )(call|make)$ #E=2,Y=1,GENERAL,ALLGOOD
^(can )(you )(help |take )(me)$ #E=2,Y=2
^(okay )(put |them )(that )(here)$ #E=2,Y=1,GENERAL,1WRONG
^(this )(is )(for )(grandmaman|maman)$ #E=2,Y=0,GENERAL,ALLGOOD
^(is )(that )(another |see )(baby)$ #E=2,Y=1,GENERAL,1WRONG
^(can |should )(you )(do )(it)$ #E=2,Y=1,GENERAL,ALLGOOD
^(what )(is )(he )(be|called|holding|interesting|missing|on|
    someone)$ #E=7,Y=2,GENERAL,2WRONG
^(where )(is )(mommy )(going|playing)$ #E=2,Y=0,GENERAL,ALLGOOD
^(gonna |where )(does )(that )(go)$ #E=2,Y=1,GENERAL,1WRONG
^(where )(are )(we )(going|playing)$ #E=2,Y=1,GENERAL,ALLGOOD
^(where )(does |likes )(it )(go)$ #E=2,Y=1,GENERAL,ALLGOOD
^(gonna |where )(is )(what )(one)$ #E=2,Y=1,GENERAL,1WRONG
^(is )(this )(the )(mommy|tractor)$ #E=2,Y=2
^(okay )(that )(is )(good|juice|nice)$ #E=3,Y=2,GENERAL,ALLGOOD
^(is )(that )(your )(friend|name|ponytail)$ #E=3,Y=2,GENERAL,
    ALLGOOD
^(do )(you )(have |love )(tea)$ #E=2,Y=1,GENERAL,ALLGOOD
^(but |touch )(who )(is )(that)$ #E=2,Y=1,GENERAL,1DISPUTABLE
^(where )(is )(my )(candy|sugar)$ #E=2,Y=2
^(try )(it )(good |on )(here)$ #E=2,Y=1,GENERAL,1DISPUTABLE
```

ACKNOWLEDGEMENTS

Ideas presented in this dissertation are result of crossover of multiple sources of intellectual and cultural influence. Some of them are enumerated in the bibliography and some of them are mentioned in acknowledgments section of my bachelor and master dissertations. Their individual names being listed elsewhere, the influence of the teachers - affiliated with Charles University in Prague, National University of Mongolia, University of Nice Sophia-Antipolis as well as of researches affiliated to 3rd Section of Ecole Pratique des Hautes Etudes (be it in Paris, Dijon, Caen, Aix en Provence or Bordeaux) - is not to be underestimated.

Also not to be underestimated is the influence of EPHE's 4th section. Although the unpredictability of life has't allowed me to pursue the "philologic track" longer than one year, I stay in the state of *reconnaissance profonde* towards prof. G-J. Pinault, J.E.M. Houben, D. Petit, A. LeMarechal, Fanny Meunier and others for showing me how "muddy" are formalisms of any theory in comparison with gems of poetry and language which have been, were, had been or simply *are* still alive.

University Paris 8 St. Dennis - renamed to Paris Lumières amidst the work on this dissertation - is also to be praised as an entity without whose support of which this dissertation would not see the light of a day. On one side its teachers, such as S. Peperkamp who introduced me into both beauties and intricacies of psycholinguistic disputes, or M.B. Jover whose courses of epistemology reignited in me a long-forgotten interest in philosophy. On another side dozens of other *fonctionnaires* who allowed me - sometimes slowly, sometimes fast, but always with success - to pursue my investigations of unknown epistemic territories.

However, EPHE and Paris 8 are just drops on the surface of structure much greater and ancient: Lutetia Parisorium, the city of Paris. Verily, if there is a three-dimensional mineral structure which is to be attributed the acknowledgment as a primary source of inspiration for this work, then it is the web which surrounds the Notre-Dame. Deeply grounded and firmly represented in my hippocampus hippocampus, Paris is indeed the city where first drafts of this dissertation had been brought to life and discussed in wines-lasting debates with Simon Carrignon, Adil elGhali, Fabien Ruggieri, Ilaria Gaudiello, Jean-Marc Thiebaut, Mary Rougeux, Yann Leger, Christophe Chavatte, Kechadi Lagha, Anne Ronsheim, Barbarka Jarkovska, Maurice & Florence Benayoun, Ivan Bigorgne, Geoffrey Tissier, Jeremy Gardent, my first student Pauline Vallies and second student Geoffrey Ventalon, Jarmila Mendelova, Louise Hearsum, Jitka Pelechova, Ophelie Monsoreau, Julienne Michele and Julie Rocton. Help coming from Kristina Poliakova, Julian Bonnyaud, Mikaela Barankova & Monique Girodroux, François Jodelet, Anh Nguyen, stream of good books coming from antiquaire Andre from bookshop on the border of rue d'Ulm and rue Claude Bernard as well as support of bouqinists Julien & Lue from Quai Voltaire were quite vital during periods when dissertation existed only *in posse* and not yet *in esse*.

But it would be unjust to praise Paris but not address some praise also *à la République toute-entière*. For without help of her institutions and her establishements - as diverse as CROUS, CAF, ANPE, Cité des Sciences, So-

ciété d'exploitation de la Tour Eiffel (!), Mairie de Paris or Campus France - it would be highly unplausible that an ordinary Bratislava boy could ever dedicate years of his life to pure science. In this regards, the role of French Ministry of Foreign Affairs, Embassy of France in Slovakia and people like Mme. Monika Saganova are of particular importance because of their assistance which ultimately allowed me to cover significant part of material needs with the scholarship of french government for doctoral studes under double supervision.

It is also thanks to Michal Oravec and Zuzana Dideková that such a double supervision got realized. By a strange coincidence of events and independently from each other they have both attracted my attention to the fact that in my own country of origin, Slovakia, there already exists a well-established, firm and intellectually rich tradition of cybernetics in general and evolutionary computation in particular. Hence I met Mr. Ivan Sekaj who was not only willing to take me under his wings, re-introduced me to education system of my own homeland, made me program my first genetic algorithm and always somehow succeeded to adopt his agenda to my needs. It is thanks to him that I had opportunity to get in contact with other "wizards from Mlynska Dolina", including prof. V. Kvasnicka or M. Popper.

None of these meetings and encounters would take place, however, if it hadn't been for one man: professor Charles Tijus. This is so because it was mainly Charles - assisted by Francois Jouen and Joelle Provasi - who kept alive the curriculum "Cognition Humain et Artificielle" at EPHE/Paris8, it was Charles who guided the direction of my Master Thesis and asides this also managed the complexities of laboratory ChART and the research platform Lutin where the germs of this dissertation have been conceived. But asides all this, it was Charles who convinced me that pursuing the path of science is worth the effort in order to subsequently give me practically absolute *liberté* in finding my own method of such pursue.

A pursue which have led me to Medienhaus of Berlin University of Arts where I have found people like prof. Alberto deCampo, Hannes Hoelzl and family, Jenny Baese, Bjorn Sickert, Joachim Sauter, Vera Garben and others. That is, people which succeed to combine humanity with professionalism in an extent I never before thought possible.

At last but not least, my ultimate "thank you" is dedicated to a woman which has transformed herself, during 6 years of doctoral studies, from a completely unknown *féerie* into a virtual acquitance into my guest into my host into my tourist guide into my friend into my love into my muse into mother of our daughter into my fiancee into my wife. It is thanks to You, Lucia, and thanks to thousands of numinous adjustments You do that our Iolanda Maitreya sleeps her green ideas peacefully and not furiously, that our house fragrances with myriads essences and this dissertation could be hereby considered as finished.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

CF	Conceptual Foundations
CI	(Category | Class | Concept) Induction
CL	Computational Linguistics
DP	Developmental Psycholinguistics
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EL	Evolutionary Linguistics
EML	Evolutionary Machine Learning
ES	Evolutionary Strategy
ET	Evolutionary Theory
FLT	Formal Language Theory
GA	Genetic Algorithm
GE	Genetic Epistemology
GI	Grammar (Induction | Inference)
GOT	Generic Optimization Toolbox)
LD	Language Development
ND	Neural Darwinism
NLP	Natural Language Processing
POS-i	Part-of-Speech Induction
POS-t	Part-of-Speech Tagging
UD	Universal Darwinism
VM	Voynich Manuscript
VSP	Vector Space Preparation

DECLARATION

I declare that this Thesis is a fruit of my own work and that all citations and references to external sources are explicitly marked.

Daniel Hromada, August 31,
2016