An Emergent Approach to Text Analysis Based on a Connectionist Model and the Web

Mario G.C.A. Cimino * and Gigliola Vaglini

Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino 1, Pisa 56122, Italy; E-Mail: g.vaglini@iet.unipi.it

***** Author to whom correspondence should be addressed; E-Mail: m.cimino@iet.unipi.it; Tel.: +39-050-2217-455; Fax: +39-050-2217-600.

Abstract: In this paper, we present a method to provide proactive assistance in text checking, based on usage relationships between words structuralized on the Web. For a given sentence, the method builds a connectionist structure of relationships between word *n*-grams. Such structure is then parameterized by means of an unsupervised and language agnostic optimization process. Finally, the method provides a representation of the sentence that allows emerging the least prominent usage-based relational patterns, helping to easily find badly-written and unpopular text. The study includes the problem statement and its characterization in the literature, as well as the proposed solving approach and some experimental use.

Keywords: natural language processing; language usage; emergent paradigm; unsupervised approach; connectionist model; web as corpus

---

1. Introduction and Motivations

Human communication processes are nowadays increasingly integrated with the Web. As a result, a huge quantity of natural language text can be instantly accessed through search engines, as a live linguistic corpus [1]. This consists of a variety of text types and styles, such as colloquial, formal, technical, scientific, medical, legal, journalistic, and so on. With respect to edited texts, web-based texts are produced in a wider variety of contexts, with different writing styles. The problem of text checking in different contexts remains to a large extent still unsolved [2]. Human language is also

influenced by evolutionary processes characterized by emergence, self-organization, collective behavior, clustering, diversification, hierarchy formation, and so on [3–6]. For this reason, new research methodologies and trends based on direct observable data have gained an increasing interest in natural language processing (NLP) [7,8]. In this paper, we present a text analysis approach that is intrinsically embodied in the Web and is based on the paradigm of *emergence*, in contrast with the classical and common paradigm of *cognitivism* [9]. In the following, we first classify NLP methods according to such paradigms, and then we provide a better characterization of our approach.

Generally speaking, in the literature there are three basic approaches to NLP, *i.e.*, symbolic, statistical and connectionist [10]. *Symbolic approaches* are based on representation of knowledge about language, derived from human introspective data. Two examples of this category are the following: (i) Rule-Based Systems, which rely on morphological/syntactic generators [11]; (ii) Semantic Networks, which are based on a structure of labeled relations and concepts [12]. Quite different are *statistical approaches*, which employ directly observable data to develop a mathematical model of linguistic phenomena [13,14]. Two examples of this category are the following: (i) Markov Models, which can predict the next symbol or word in a sequence [15,16]; (ii) Language Usage Patterns, in which NLP expressions are analyzed by means of surveys for performing statistical inference [2]. By contrast, in *connectionist approaches* a model is a network of interconnected simple processing units with knowledge embodied in the weights of the connections. Connections reflect local structural relationships that can result in dynamic global behavior. Similarly to statistical approaches, connectionist approaches develop models from observable data. However, with connectionist systems linguistic models are harder to observe, because the architectures are less constrained than statistical ones, so as to allow *emergent* phenomena [9].

Both symbolic and statistical approaches belong to the cognitivist paradigm [9]. In this paradigm, the system is a descriptive product of a human designer, whose knowledge has to be explicitly formulated for a representational system of symbolic information processing. This designer-dependent representation biases the system, and constrains it to a consequence of the cognitive analysis of human activity. Indeed, it is well known that symbolic systems are highly context-dependent, neither scalable nor manageable [17], ineffectual in optimizing both grammar coverage and resultant ambiguity [17]. With respect to symbolic systems, statistical systems are more robust in the face of noisy and unexpected inputs, allowing broader coverage and being more adaptive [10]. Actually, every use of statistics is based upon a symbolic model, and statistics alone is not adequate for NLP [2,10]. In contrast, connectionist systems can exhibit higher flexibility, by dynamically acquiring appropriate behavior on the given input, so as to be more robust and fault tolerant [10].

The connectionist approach discussed in this paper takes inspiration from the *emergent paradigm* [9], which reflects the dynamic sociological characteristics of natural languages. The underlying idea is that simple mechanisms, inspired by basic human linguistic capabilities [16], can lead to an emergent collective behavior, representing an implicit structure of the sentence in terms of relationships between words. With this approach, the most important consideration in the modeling is that global (*i.e.*, language) -level relationships between words must not be explicitly modeled, neither in logical nor mathematical terms. Such relationships must be kept *embodied* in the corpus [9]. Indeed, in contrast with a cognitivist system, which does not need to be embodied, an emergent system is dependent on the physical platform in which it is implemented, *i.e.*, the platform in which the corpus itself resides [9].

When using the Web as a corpus, representativeness and correctness are two important topics of debate [1,18,19]. With regard to representativeness, let us consider some typical events of human conversation and their availability in both web-based and conventional text. *Production and reception*: many conversations have one speaker and one hearer; this one-to-one conversation is largely available on the Web; in contrast, many conventional text have one writer and many readers, e.g., a Times newspaper article. *Speech and text*: there are orders of magnitude more speech events than writing events; web-based messaging is very close to speech events; in contrast, most conventional corpus research has tended to focus on text production rather than on speech production. *Background language*: rumors and murmurs are conversational events greatly available on social networks; in contrast, these kinds of events are poorly covered by conventional text. *Copying*: in the text domain, copyright, ownership and plagiarism restrict cut-and-paste authorship, whereas in the Web domain the open access paradigm enables new language production events.

With regard to correctness, a fundamental paradigm shift has been occurring since the introduction of the Web as a corpus. In contrast to paper-based, copy-edited published texts, web-based texts are produced by a large variety of authors, cheaply and rapidly with little concerns for formal correctness [1]. For instance, a Google search for "I beleave", "I beleive", and "I believe" gives 257,000, 3,440,000, and 278,000,000 hits, respectively. Hence, all the "erroneous" forms appear, but much less often than the "correct" forms. From the formal standpoint, the Web is a dirty corpus, but expected usage is much more frequent than what might be considered noise. Actually, a language is made of a core of lexis, grammar, constructions, plus a wide array of sublanguages, used in each of a myriad of human activities. In the last decade, an extensive literature on sophisticated mathematical model for word frequency distributions has been produced with the aim of modeling sublanguage mixtures [1,20].

Let us consider a simple positive feedback: for a given sentence, the more occurrences of the sentence in a corpus, the more correctness of the sentence [21]. Here, the *open-world assumption* is considered: any phraseology that is used in some sublanguage events of human conversation can be positively assessed [20]. However, it is unlikely that many occurrences of the same sentence are found in a corpus [1]. Moreover, for an incorrect sentence it should be important to show which part of the sentence is actually incorrect. Hence, a structural analysis of the sentence able to allow emerging relationships between words should be considered. Here, we emphasize that this analysis should be performed at the syntagmatic level, by identifying and rating elementary segments within the text (syntagms) [22]. Nevertheless, the number of occurrences of a text segment is strongly affected by the usage of its terms. For instance, unfamiliar proper nouns and unusual numbers may drastically limit the number of occurrences of a segment. Hence, some transformations of segments should be taken into account, to allow substitution of terms within the same category that does not affect the structural relationships.

In our approach, we avoid identifying codes, rules or constraints that underlie the production and interpretation of text. For this reason, our method could be applied, with no changes, to many other languages that have enough available *n*-grams on the Web. The fundamental assumption of our grammarless approach is that the strength of word relationships can arise via a structural disassembly process of the sentence, upon language agnostic operators such as segmentation and substitution. This process is fundamental so as to allow the sublanguages knowledge to be kept embodied in the corpus.

To avoid an explicit representation, words relationships are represented in a connectionist model [17,23,24], whose weights are trained via an unsupervised optimization process. Here, clustering is essential to identify atypical and misused parts, structurally opposed to commonly used parts. Finally, from the connectionist model, an output is derived so as to provide a visual representation [25] of the sentence able to give the writer an informative insight of the text usage.

The paper is organized as follows. Section 2 covers the related work on open-world approaches to textual analysis. In Section 3, we introduce the problem formulation. Section 4 is devoted to the connectionist model and its components. Section 5 describes the determination of weights of the connections. Section 6 is focused on experimental results. Section 7 draws some conclusions and future works.

2. Open-World Approaches to Textual Analysis: Related Work

To the best of our knowledge, no work has been done in the field of text analysis using a connectionist model and the Web. However, there are a number of research projects that pursue textual analysis tasks using the Web as a corpus. In this section, we intend to characterize and present such *open-world* approaches with the aim of providing a landscape of the current methodologies.

In the *closed-world* assumption, any linguistic analysis that cannot be generated by the grammar is assumed to be ungrammatical. In contrast, statistical parsers are considerably more open-world. For example, unknown words do not present a problem for statistical parsers. A possible approach to produce more open grammar-based approach is to relax the interpretation of constraints in the grammar. For instance, rules can be interpreted as soft constraints that penalize analyses in which they fail. However, any option that makes the grammar-based approach open-world requires a very higher computational effort, and needs parsing algorithms capable of handling massive ambiguity [20].

Grammar-based approaches model explicit linguistic knowledge that is closer to meaning. Indeed, grammar-based analyses explicitly represent predicate-argument structure. However, predicate-argument structure can be also recovered using statistical methods [26]. Grammar-based approaches are also often described as more linguistically based, while statistical approaches are viewed as less linguistically informed. However, this difference between the two approaches is misleading [20], because there are only different ways of modeling linguistic knowledge in the two approaches. Indeed, in the grammar-based approach linguists explicitly write the grammars, while in statistical approaches linguists annotate the corpora with syntactic parses. Hence, linguistic knowledge plays a central role in both approaches. While many features used in statistical parsers do not correspond to explicit linguistic constraints, such features encode psycolinguistic preferences and aspects of world knowledge. Hence, from a high-level perspective, the grammar-based and the statistical approaches view parsing fundamentally in the same way, namely as a specialized kind of inference problem [20].

A direct comparison with our system in terms of result is not currently feasible, due to functional, architectural and structural differences with the open-world approaches to textual analysis available in the literature.

From a *functional* standpoint, the research field of open-world approaches to text correction is characterized by a variety of specialized NLP sub-tasks. Examples of NLP tasks are: real-world error

correction; near-synonym choice; preposition choice; adjective correction; adjective ordering; context-sensitive spelling correction; part-of-speech tagging; word sense disambiguation; noun countability detection; language-specific grammatical error correction made by native-language-specific people, and so on [27]. Common examples of application of statistical NLP are: the classification of a period as end-of-sentence; the classification of a word into its part-of-speech class; the classification of a link between words as a true dependency. In contrast, our system does not model linguistic sub-tasks.

From an *architectural* standpoint, for each aforementioned NLP sub-task linguistic knowledge is injected in the system through specific algorithms, parameters, and training data. Most tasks of statistical NLP methods to text correction are classification problems tackled via machine learning methods. Classifiers can logically be trained only on specific linguistic problems and on a selected data set. Training process leads to scalability issues when applied to complex problems or to large training sets without guidance. For this reason, web-based NLP models are typically supervised models using annotated training data, or unsupervised models which rely on external resources such as taxonomies to strengthen results. In contrast, our system does not adopt some form of linguistic training or some form of linguistic supervision.

From a *structural* standpoint, with a linguistically informed approach there is a dualist distinction between computational processes and data structures. In contrast, our emergent system is characterized by fine-grained coupling between behavioral model and environment. Indeed, web data organization is a structural part of the algorithm, and data output is comprehensive and visually well integrated with the human perception (embodiment).

More specifically, in the remainder of the section we summarize the open-world approaches to text detection and correction relevant with respect to our work.

In [14], the authors present a method for correcting real-world spelling errors, *i.e.*, words that occur when a user mistakenly types a correctly spelled word when another was intended. The method first determines some probable candidates and then finds the best one among them, by considering a string similarity function and a frequency value function. The string similarity function is based on a modified version of the Longest Common Subsequence (LCS) measure. To find candidate words of the word having spelling error, the Google Web 1T *n*-gram data set is used.

An unsupervised statistical method for correcting preposition errors is proposed in [19]. More specifically, the task is to find the best preposition from a set of candidates that could fill in the gap in an input text. The first step is to categorize an *n*-gram type based on the position of the gap in the Google *n*-gram data set, *n* ranging from 5 to 2. In the second step, the frequency of the *n*-gram is determined, and then the best choice preposition is established.

In [28] the authors propose a way of using web counts for some tasks of lexical disambiguation, such as part-of-speech tagging, spelling correction, and word sense disambiguation. The method extracts the context surrounding a pronoun (called *context patterns*) and determines which other words (called *pattern fillers*) can take the place of the pronoun in the context. Pattern fillers are gathered from a large collection of *n*-gram frequencies. Given the *n*-gram counts of pattern fillers, in the supervised version of the method, a labeled set of training examples is used to train a classifier that optimally weights the counts according to different criteria. In the unsupervised version, a score is produced for each candidate by summing the (un-weighted) counts of all context patterns.

A method for detecting grammatical and lexical English errors made by Japanese is proposed in [29]. The method is based on a corpus data, which includes error tags that are labeled with the learners' errors. Error tags contain different types of information, *i.e.*, the part of speech, the grammatical/lexical system, and the corrected form. By referring to information on the corrected form, the system is able to convert erroneous parts into corrected equivalents. More specifically, errors are first divided into two groups, *i.e.*, the omission-type error and the replacement-type error. The former is detected by estimating whether or not a necessary word is missing in front of each word, whereas the latter is detected by estimating whether or not each word should be deleted or replaced with another word. To estimate the probability distributions of data the Maximum Entropy (ME) model is used. Finally, the category with maximum probability is selected as the correct category.

In [30] a method for detecting and correcting spelling errors is proposed, by identifying tokens that are semantically unrelated to their context and are spelling variations of words that would be related to the context. Relatedness to the context is determined by a measure of semantic distance. The authors experimented different measures of semantic relatedness, all of which rely on a WordNet-like hierarchical thesaurus as their lexical resource.

A multi-level feature based framework for spelling correction is proposed in [31]. The system employs machine learning techniques and a number of features from the character level, phonetic level, word level, syntax level, and semantic level. These levels are evaluated by a Support Vector Machine (SVM) to predict the correct candidate. The method allows correcting both non-word errors and real-world errors simultaneously using the same feature extraction techniques. The method is not confined to correct only words from precompiled lists of confused words.

In [32], the authors analyze the advantages and limitations of the trigrams method, a statistical approach that uses word-trigram probabilities. Conceptually, the basic method follows the rule: if the trigram-derived probability of an observed sentence is lower than that of any sentence obtained by replacing one of the words with a spelling variation, then the original is supposed to be an error and the variation corresponds to what the user intended. The authors present new versions of this algorithm that use fixed-length windows, designed so that the results can be compared with those of other methods.

An efficient hybrid spell checking methodology is proposed in [33]. The methodology is based upon phonetic matching, supervised learning, and associative matching in a neural system. The approach is aimed at isolated word error correction. It maps character onto binary vectors and two storage-efficient binary matrices that represent the lexicon. The system is not language-specific and then it can be used with other languages, by adapting the phonetic codes and transformation rules.

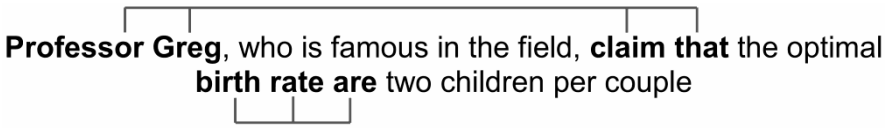
3. Problem Formulation

Our system aims at providing a continuous-valued representation of word relationships in a given sentence. Figure 1 shows a sample sentence with some relationships (connections) between words. Here, subsequences of words, *i.e.*, word *n*-grams, involved in each connection are represented in boldface.

Some general properties of subsequences are the following: (i) subsequences can be made of non-contiguous words, as represented in the first *n*-gram; (ii) subsequences can be overlapped; (iii) a

suitable number of subsequences can be generated so as to cover all the words in the sentence; (iv) a subsequence does not usually correspond to a clause, since a grammarless approach is used.

Figure 1. Example sentence with some arcs showing dependencies between words.



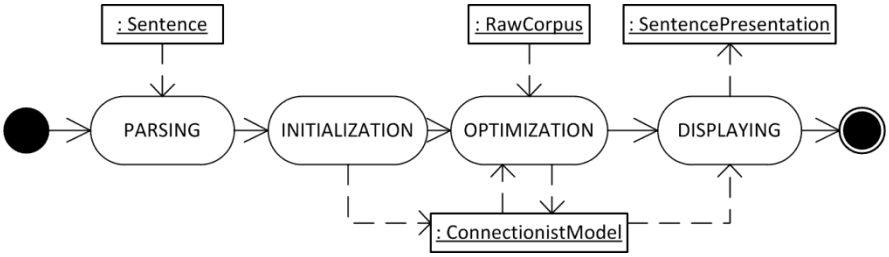
A sentence with the corresponding subsequences can be represented as a connectionist model. Each word is represented by a node of the network, and the connections between nodes represent word relationships. Weak (or strong) connections model weak (or strong) relationships between its words. Connections strength can be based on the usage of their subsequences on the Web.

In general, different segmentations of a sentence in subsequences are possible. Hence, a suitable optimization method should be able to identify the better segmentation so as to emphasize subsequences with very low usage.

Finally, a suitable displaying method should provide an intuitive manner of expressing the relevant information owned by the connectionist model.

Figure 2 shows an UML activity diagram of the macro activities of our approach to text analysis. Here, activities (represented by oval shapes) are connected via control flow (solid arrow), whereas input/output data object (rectangles) are connected via data flow (dashed arrow).

Figure 2. Overall activities involved in our emergent approach to text analysis.



At the beginning, the sentence is *parsed* and then converted into an initial *connectionist model* instance. The sentence is completely broken up into (overlapped) segments by a *segmentation* operator. Afterwards, the *connectionist model* instance goes through an *optimization* process, which optimizes the connections by using the usage information available in the *raw corpus* instance, *i.e.* the Web. Finally, *connectionist model* information is *displayed*, *i.e.*, transferred to a visual representation of the sentence, namely a *sentence presentation* instance [25,34].

More specifically, Figure 3 shows the macro activities of the optimization process. First, a *segmentation* of the sentence is performed, producing a series of *n*-grams of the sentence itself. Then, one of two possible operators is applied, namely *generalization* or *commutation*. The former is an operator that employs the class of a specific word in place of the world itself, whereas the latter is an operator which substitutes a word with another more popular word which is structurally similar. Afterwards, the *usage* of each *n*-gram in the corpus is rated. Finally, the *n*-grams with the lowest usage

are determined. In order to find the best setting, all these activities may be carried out a number of times, as represented by the loop in the Figure.

Figure 3. Macro activities of the optimization process.

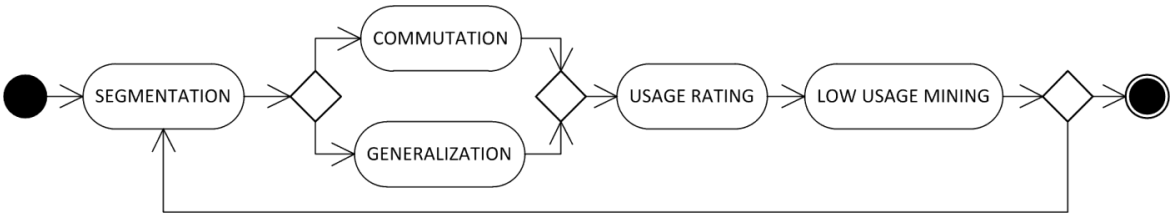
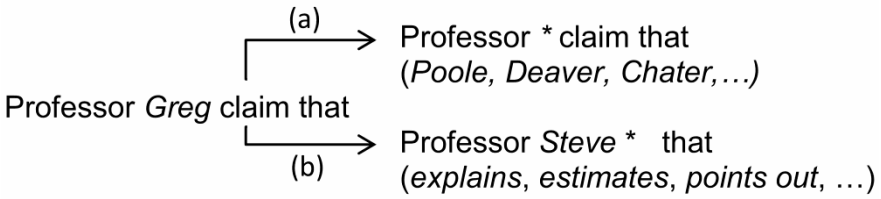


Figure 4a shows an example of generalization, in which the individual name "Greg" is replaced with any other individual name that can be found in the Corpus by using the wildcard. Figure 4b shows an example of commutation, in which the individual name "Greg" is replaced with the more popular name "Steve" and the word "claim" is generalized. Each of these alternatives affects the *usage rating* of the phrase, and allows a better robustness of the *optimization* process. For example, without the generalization operator, usage rating may be strained by an unpopular word.

Figure 4. An example of generalization (**a**) and commutation (**b**) operators.



In the next section, we introduce some definitions to formalize our method.

4. The Connectionist Model and Its Components

4.1. Input Sentence and Operators

Let us consider an input sentence *G*, with *n* items g_i after tokenization:

$$G = (g_1, ..., g_n),\ \mathrm{G(i)} \triangleq \mathrm{g}_i \tag{1}$$

Tokenization is case sensitive, and makes a different item for each word and each punctuation mark. All words are supposed to be correctly spelled (Search engines already figure out possible misspelling and their likely correct spellings, using a character n-gram models. For example, the "did you mean" generator offered by Google Inc.). Sentence ends with a full stop, an exclamation mark, a question mark, or a semicolon:

$$\mathrm{g}_n \in \{.\ !\ ?\ ;\} \tag{2}$$

The *segmentation* operator divides an *n*-gram into partially overlapping *s*-grams (segments) with $s > 1$. The extent of overlapping is established by the parameter *o*, *i.e.*, the number of items common to any subsequent segments:

$$\text{Segment}(G, s, o) \triangleq ((g_1, ..., g_s), (g_{s-o+1}, ..., g_{s-o+1+s}), ..., (g_{n-k+1}, ..., g_n)),\ o < s < n \quad (3)$$

Depending on *n*, the length of the last segment (*k*) can be either equal to $s + 1$ or lower than *s*.

The *generalization* operator substitutes an *n*-gram with a defined set of possible equivalent items, according to an equivalence type (θ):

$$\text{Generalize}(G) \triangleq \{G' | G' \equiv_\theta G\} \quad (4)$$

Table 1 shows some important examples of such operator.

Table 1. Some examples of the *generalization* operator.

	Operation	**Result**
(i)	Generalize("Greg")	Any individual name
(ii)	Generalize("18754")	Any number
(iii)	Generalize(",who is famous in the field,")	Any *n*-gram between commas, or nothing.

4.2. Search Engines and Hit Counts

The default means of access to the Web is a search engine. In particular, our method uses the hit counts and examines a limited number of snippets, *i.e.*, short descriptions available in results pages. Hence, the method does not require an expensive downloading of actual text for analysis. Snippets allow inspecting results so as to filter a percentage of irrelevant matches. Unfortunately, search engines were not designed for NLP tasks and the reported hit counts are subject to inaccuracy [1,7]. For instance, search is not case sensitive, it pays no attention to punctuation marks, and word considered adjacent to each other could actually lie in different sentences of paragraphs. Hence, according to [2] we define some basic concepts.

Given an *n*-gram *G*, we use $\|G\|_\Theta$ to denote the *raw usage* as the number of Web pages (hit counts) containing *G* found by the search engine Θ. The *precise usage* $\|G\|_{\Theta,\pi}$ represents the raw usage excluding a proportion due to inaccurate results found in the hit counts according to a precision parameter π. More specifically:

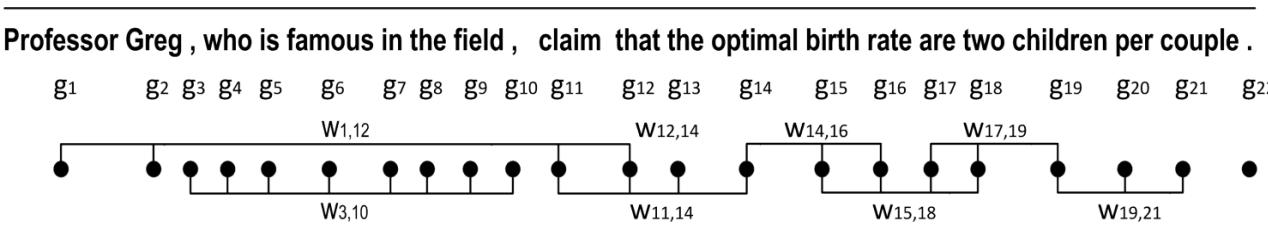
$$\|G\|_{\Theta,\pi} = \|G\|_\Theta \cdot \sigma / \sigma_\pi \quad (5)$$

where σ_π is the number of snippets found by parsing a number of pages equal to π, and σ is the number of snippet with an exact match by considering also punctuation marks, case-sensitivity, adjacency, generalization (4).

4.3. The Connectionist Structure

Figure 5 shows a connectionist model related to the example sentence of Figure 1. Each item g_i of the sentence (words and punctuation marks) is represented by an individual unit u_i (localist model). The number of units is then prefixed for a given input sentence, whereas the number of connections and their weights are determined by the optimization algorithm.

Figure 5. A connectionist model with dependencies between words.



The basic type of relationship in a sentence concerns a totally joined *n*-gram, (g_i,…, g_j) [17] with arcs $W_{i,j}$, which is modeled by a connection between units u_i,…, u_j, with strength $w_{k,h}$. An example of this relationship is $W_{14,16}$ in Figure 5. Note that such *n*-grams in general do not correspond to clauses or other grammatical concepts that could be labeled: a connection could be virtually established on any subsequence. The optimization procedure limits the number of connections. Another type of relationship concerns a partially joined *n*-gram. An example of this relationship is $W_{1,12}$ in Figure 5, involving g_1, g_2, g_{11}, and g_{12} only.

For a network with *N* nodes, the *k*-th output, $k = 1,.., N$, is the following:

$$out(u_k) = \frac{1}{p_k} \sum_h (w_{k,h})^2 \tag{6}$$

where $w_{k,h}$ are the weights corresponding to the p_k connections related to the node u_k.

4.4. The Visual Output of the Network

The output provided by the network is visually represented by using size and color of the text [25,34]. The font size represents the usage of contiguous *n*-grams, whereas the foreground text color represents the usage of non-contiguous *n*-grams. More specifically, given $t_{\min}$ and $t_{\max}$ the minimum and maximum font size, respectively, the average font size of the *k*-th word belonging to a contiguous *n*-gram is the following:

$$t_k = t_{\min} + (t_{\max} - t_{\min}) \cdot out(u_k) \tag{7}$$

whereas the average color of the *k*-th word belonging to a non-contiguous *n*-gram is the following, expressed in terms of red (R), green (G), and blue (B) coordinates:

$$\left[R_k,\ G_k,\ B_k\right] = \begin{cases} \left[1,\ 2\cdot out(u_k),\ 0\right] & if\ \ out(u_k) \le 0.5 \\ \left[1,\ 1,\ 2\cdot out(u_k) - 1\right] & if\ \ out(u_k) \ge 0.5 \end{cases} \tag{8}$$

Figure 6 shows a large range of achievable colors with the corresponding value of $out(u_k)$, *i.e.*, Y = (G + B)/2.

Figure 7 shows an example of visual output, where there are two *n*-grams with low usage, *i.e.*, "Professor Greg claim that" and "Rate are 2", non-contiguous and contiguous respectively. Here, it can be also noted that letters of a single word have different sizes and colors, so as to have soft style transitions. Indeed, the size and colors computed by the above formulas are average values for each word, which are linearly spread from a word to another.

Figure 6. Colorization of a value Y between 0 and 1: Y = (G + B)/2.

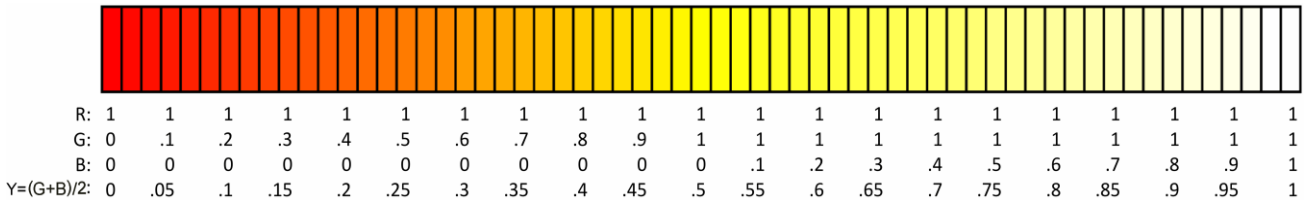
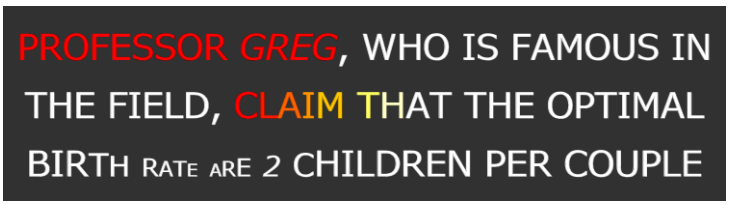


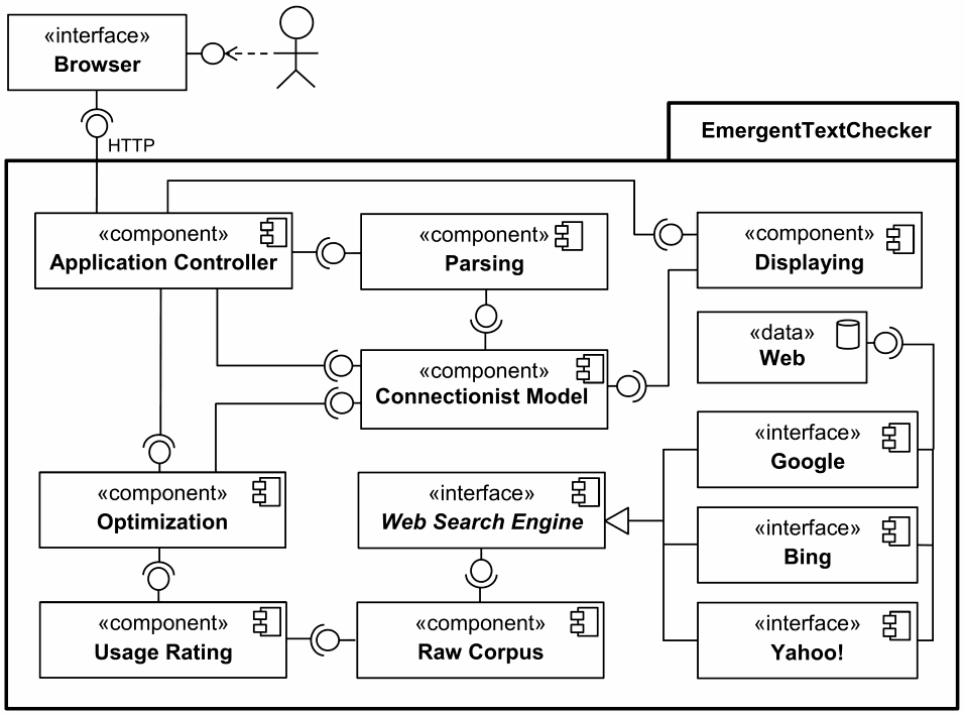
Figure 7. Example of visual output.



4.5. Overall Components of the System

Figure 8 shows how the overall components of the system are wired together, via a UML component diagram. The client-side is made of a web-browser interface. The package *Emergent Text Checker* contains all the server-side components. On the server-side, an *Application Controller* component manages all the communications from and to the client-side, as well as triggers the other components. More specifically the main components are the following: (i) the *Parsing* and *Displaying* components, which manage the input and the output sentences, respectively; (ii) the *Connectionist Model* component, which is responsible for managing the connections between words; (iii) the *Optimization* component, which is able to optimize the *Connectionist Model* on the basis of the *Usage Rating* component; (iv) the *Raw Corpus* component, which can be realized thanks to the use of the *Web Search Engine* components. The latter can be implemented with many alternatives, *i.e.*, Google, Bing, Yahoo!, or with an aggregation of their results.

Figure 8. Overall system components.



5. The Determination of the Weights

In this section, we elaborate on the determination of the weights of the connectionist model. Weights are mainly established via an optimization procedure, which aims at separating low usage from normal/ high usage. For a single optimization process, different segmentations of the sentence are possible. For each segmentation process, the precise usage of each *n*-gram is calculated.

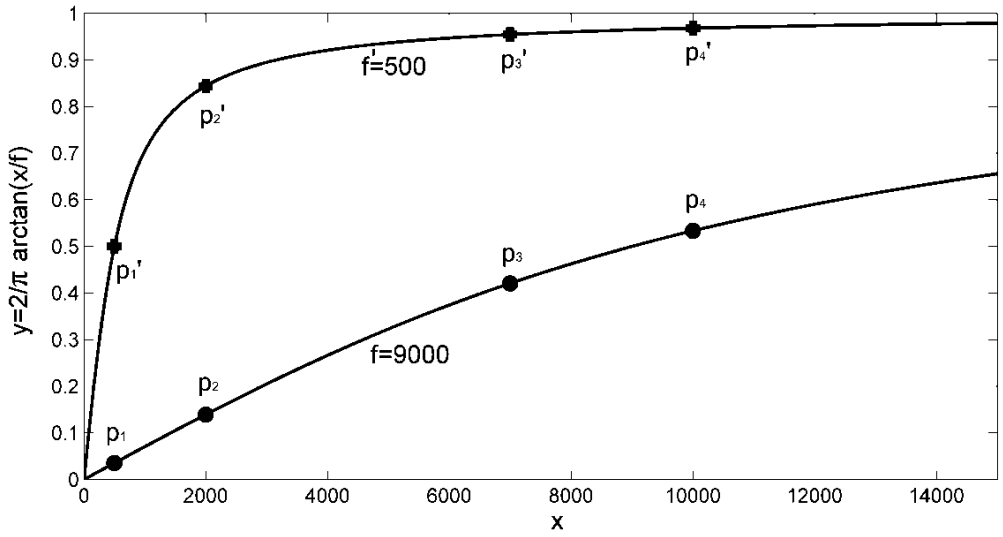
Usage values of the connections are divided into categories so that usage values in the same category are as similar as possible, and usage values in different categories are as dissimilar as possible. Further, each usage value can belong to more than one category. This soft clustering process is used to optimize the weights of the connections in the network.

The optimization process aims at discovering low usage segments in the sentence. For this reason we adopt the following proximity function, which tends to zero as x_1 and x_2 tend to infinity:

$$d(x_1, x_2) = \frac{2}{\pi}\left|\arctan(x_1 / f) - \arctan(x_2 / f)\right| \tag{9}$$

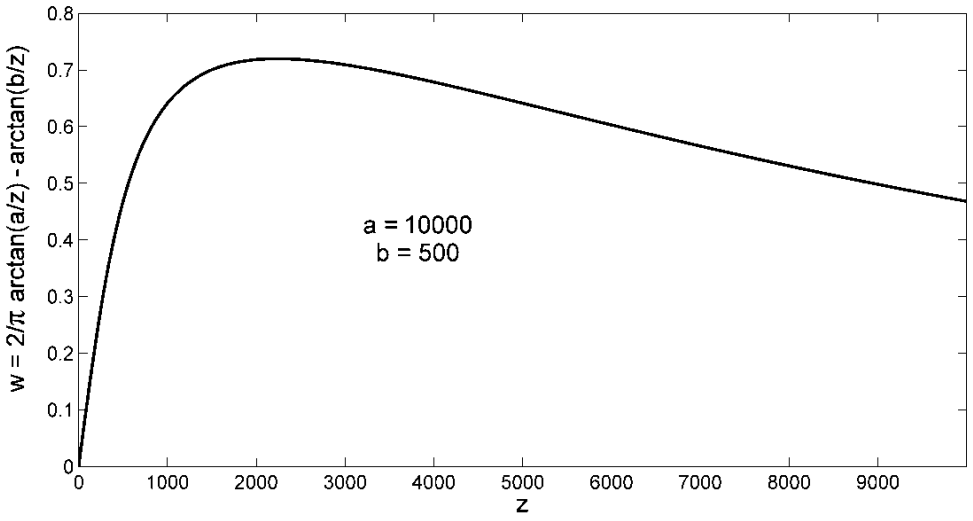
where f is a scaling factor which is automatically adapted. More specifically, Figure 9 shows two simple scenarios of proximity space $y = arctan(x/f)$, corresponding to two different curves with different values of the scaling factor f. On both curves, the same precise usage values are considered, *i.e.*, $x_1, \ldots, x_4$. With the lower scaling factor ($f' = 500$), high usage values are considered very similar in the proximity space, whereas low usage values are considered very dissimilar. However, a very low scaling factor would consider all usage values almost identical and equal to 1 in the proximity space. With the higher scaling factor ($f = 9000$), usage values in the proximity space y are almost linearly connected with the source space.

Figure 9. Two simple scenarios with the adopted proximity space.



In our approach, the scaling factor is automatically adapted by maximizing the proximity between the minimum and the maximum usages in the sentence, e.g., y_1 and y_4 in Figure 9. Figure 10 shows an example of differential proximity space $w = arctan(a/z) - arctan(b/z)$, with $a > b$. The example clearly shows that there is a unique global maximum of the proximity between a and b, that can be easily found by means of fundamentals of mathematical analysis. In conclusion, by using (9) with the adaptation of the scaling factor, high usage values are all considered similar, whereas differences between low usage values are sensed.

Figure 10. A scenario of differential proximity space.



We adopted an implementation of a soft clustering process known as Fuzzy C-Means (FCM), with a simple iterative scheme and good convergence properties [35]. The algorithm categorizes a set of data points $\mathbf{U} = (\mathbf{u}_1,...,\mathbf{u}_n)$ finding D cluster centers $\mathbf{C} = (\mathbf{c}_1,...,\mathbf{c}_D)$ as prototypes and the fuzzy membership degrees $\mathbf{N}_i = (\mu_{i1},...,\mu_{iD})$ of each data point $\mathbf{u}_i$ to the cluster centers, under the constraint $\sum_{j=1}^{D} \mu_{ij} = 1$. The FCM algorithm introduces fuzzy logic with respect to the well-known K-Means (or Hard C-Means, HCM) clustering algorithm. The two algorithms are basically similar in design. The

latter forces data points to belong exclusively to one category, whereas the former allows them to belong to multiple clusters with varying degrees of membership. Such degrees are crucial for measuring the quality of the process as well as for the determination of the connection weights, and then the fuzzy character of the clustering can be considered a requirement of our approach.

There is a plethora of fuzzy clustering methods available in the literature [36]. For instance, the Fuzzy Self-Organizing Map (Fuzzy SOM) can be taken into consideration, as well as many other FCM derivatives. The most of them are iterative methods. Moreover, some of them are more robust to outliers, and less sensitive to the initial conditions. However, in our study most performance-related problems on the clustering are mitigated, because the clustering is made on a mono-dimensional space, with a proximity function that facilitates the granulation process, with a limited number of points and of clusters. Thus, the clustering process converges very quickly, in a very few iterations. We adopted the basic FCM version as it has been used very successfully in many applications, having a simple iterative scheme and good convergence properties.

The FCM algorithm minimizes an objective function representing a clusters compactness measure, by iteratively improving fuzzy membership degrees until no further improvement is possible. More specifically, the cluster centers are computed as the weighted average of all data points, *i.e.*, $c_h = \sum_{i=1}^{n} \mu_{ih}^m \cdot u_i \Big/ \sum_{i=1}^{n} \mu_{ih}^m$, whereas the fuzzy membership degrees are computed as follows:

$$\mu_{ih} = 1 \Big/ \sum_{\substack{j=1 \\ j \neq i}}^{D} \left(\frac{d(u_i, c_h)^2}{d(u_i, c_j)^2} \right)^{1/(m-1)} \tag{10}$$

where d is a proximity function and $m > 1$ is a parameter called *fuzziness*. The choice of the proximity function determines the success of a clustering algorithm on the specific application domain [37]. As a proximity function, we adopted formula (9), which facilitates the granulation process. FCM approaches HCM when m is approaching 1. The larger the value of m (up to infinity), the larger the similarity of the clusters. The parameter is usually set to 2. We adopted this value since its effect is marginal in our system.

The FCM method requires also the number of categories D as input. Different fuzzy partitions are obtained with different number of categories. Thus, a cluster validity index is required to validate each of the fuzzy partitions and to establish the optimal partition [38], *i.e.*, the optimal number of categories. The FCM validation procedure used to determine the optimal number of clusters is made of the following steps:

(i) initialize the parameters of the FCM except for the number of clusters, *D*;
(ii) execute the FCM algorithm for different values of D, ranging from 2 to a maximum, established in the design stage or at runtime;
(iii) compute the validity index for each partition provided by step (ii);
(iv) choose the optimal partition and the optimal number of categories according to the validity index.

To find the optimal number of categories, we adopted the Xie-Beni validity index, which optimizes compactness and separation of categories [39]:

$$P = \frac{\sum_{i=1}^{n}\sum_{j=1}^{D}\mu_{ij}^{m}\cdot d(u_i,c_j)^2}{n\cdot\min\limits_{i\neq j} d(c_i,c_j)} \tag{11}$$

where the numerator and the denominator indicate compactness and separation, respectively. Thus, the best partition corresponds to the minimum value of P.

In conclusion, the overall optimization process can be summarized as follows. First, a segmentation of the sentence into subsequences is performed. Second, the usage values (points) of each subsequence in the Web are computed. Third, a number of clusters is chosen. Fourth, point coefficients are assigned randomly for each cluster. Fifth, the centroid of each cluster is computed. Sixth, point coefficients are computed for each cluster. Seventh, go to the fifth step, if there is no convergence in coefficients. Eighth, the Xie-Beni index of clusters is computed. Ninth, go to the third step if a new number of clusters should be assessed. Tenth, provide the coefficients of the clustering process related to the best Xie-Beni index. Eleventh, coefficients are employed to assign the weights of the network.

More formally, the optimization algorithm can be defined as follows.

Algorithm: optimization of the weights in the connectionist model

---

01: $G \leftarrow$ Tokenize(*input sentence*);

02: $G' \equiv (G^{(1)}, G^{(2)},\ldots, G^{(n)}) \leftarrow$ Segment(G,s,o);

03: $U \equiv (u_1,\ldots, u_n) \leftarrow (\|G\|_{\Theta,\pi}^{(1)}, \|G\|_{\Theta,\pi}^{(2)},\ldots, \|G\|_{\Theta,\pi}^{(n)})$;

04: $C_{opt} \leftarrow \varnothing$; $M_{opt} \leftarrow \varnothing$; $P_{opt} = \infty$;

05: **for** D = 2 to 5 **do**

06: $t \leftarrow 0$;

07: Initialize $\mu_{ih} \in [0,1]$, $1 \le i \le$ n, $1 \le h \le D$ (categories);

08: **do**

09: $c_h^{(t)} \leftarrow \dfrac{\sum_{i=1}^{n}\mu_{ih}^{m}\cdot u_i}{\sum_{i=1}^{n}\mu_{ih}^{m}}$, $1 \le h \le$ D;

10: $\mu_{ih}^{(t)} \leftarrow \dfrac{1}{\sum\limits_{\substack{j=1\\ j\neq i}}^{D}\left(\dfrac{d(u_i,c_h)^2}{d(u_i,c_j)^2}\right)^{1/(m-1)}}$, $1 \le h \le$ D, $1 \le i \le n$;

11: $t \leftarrow t$+1;

12: **while** $\max\limits_{ih}\left|\mu_{ih}^{(t-1)} - \mu_{ih}^{(t)}\right| \le \varepsilon$;

13: $P \leftarrow \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{D}\mu_{ij}^{m}\cdot d(u_i,c_j)^2}{n\cdot\min\limits_{i\neq j} d(c_i,c_j)}$;

14: **If** $P \le P_{opt}$

15: $P_{opt} \leftarrow P$; $C_{opt} \leftarrow \{c_h\}$; $M_{opt} \leftarrow \{\mu_{ih}\}$;

16: **end if**

17: **end for**

The result of this optimization process is made of: (i) the usage categories C_{opt}; (ii) the membership degrees of each segment usage to all categories, M_{opt}. Let us assume that the lowest category is identified by $h = 1$. Hence, we take $w_{k,h} \equiv 1 - \mu_{1j}$ in order to discover atypical, misused and outdated segments in the sentence.

Table 2 summarizes the parameters of the system, together with their typical values. Such values have been derived by maximizing the performance of the system over a subset of the sample sentences used in the experimental results (Section 6).

Table 2. Parameters of the algorithm and their typical values.

Parameter	Description	Value	Reference
Θ	Search engine	google, bing, yahoo, all, random	Section 4.2
π	Number of snippet pages to parse	from 2 to 5, to improve precision	Section 4.2
o	Allowed overlapping *n*-grams	1, 2	Section 4.1
l_{MIN}, l_{MAX}	Minimum and maximum allowed length of *n*-grams.	3, 4	Section 4.1
f	Initial threshold of low usage	3,000,00	Section 5
t_{MIN}, t_{MAX}	Minimum and maximum font size	1,030	Section 5

6. Experimental Results

In order to test the effectiveness of the system, a collection of 80 sentences have been derived from the British National Corpus (BNC) [40]. More specifically, the extraction criterion was the following. First, the following list of the most frequent English word has been derived: *time*, *year*, *people*, *way*, *man*, *day*, *thing*, *child* [41]. Second, word pairs in the list have been used as a search criterion to find a collection of 30 sentences. Third, a new collection of 50 sentences has been produced by introducing mistakes in the first collection of sentences, and thus having 80 total sentences.

In order to measure the system performance, let us consider the system as a classifier whose results (expectation) are compared under test with trusted external judgments (observation). A correct result (*true positive*) is then an atypical subsequence discovered in the sentence, whereas a correct absence of result (*true negative*) is a good sentence where no atypical subsequence has been discovered, *i.e.*, the lowest usage category is empty.(Actually, the lowest usage category contains the zero usage value by default, and then this condition from a technical standpoint means that the category contains the zero usage value only) Hence, the terms *positive* (the sentence is somewhere atypical) and *negative* (the sentence is good) refer to the *expectation*, whereas the terms *true* and *false* refer to whether that expectation corresponds to the observation.

Figure 11 shows some examples of successful application of our system. Here, each black rectangle is a visual output of an input sentence. On the left side of the Figure, original BNC sentences are presented. All these sentences are correct from a grammatical standpoint, and then no atypical subsequences are available in the sentences. Hence, all cases on the left are true negatives. On the right side of the Figure, the same sentence of the left side is presented with some grammatical mistake, so as to have some atypical subsequence. In all cases, the system correctly identified the atypical segment. Hence, all cases on the right are true positive. It is worth noting, on the right of Figure 11k and Figure 11 (l), two examples of colored non-contiguous subsequences.

Table 3 shows some values related to the sentences of Figure 11c. Here, it can be easily noticed that the atypical subsequence (represented in boldface) is characterized by a weight value $w_{k,h}$ lower than the corresponding scaling factor *f*.

Table 3. Values related to the sentences of Figure 11c.

That was one man he wanted people to grieve for.		
n. of subsequences: 4		
n. of clusters: 2		
Xie-Beni index: 0.0000070033		
f = 29,851		
n-grams	$\|G\|_{\Theta,\pi}$	$w_{k,h}$
that was one	888,034,526	0.9999888
one man he	22,619,593	0.9999936
he wanted people to	3,159,405	0.9999929
to grieve for	2,896,900	0.9999896

That was one man he wanted people to grieving for.		
n. of subsequences: 5		
n. of clusters: 2		
Xie-Beni index: 0.000032159		
f = 30,249		
n-grams	$\|G\|_{\Theta,\pi}$	$w_{k,h}$
that was one	909,886,666	0.9999704
one man he	25,481,131	0.9999779
he wanted people	3,272,179	0.9999997
people to grieving	**452**	**0.0000233**
grieving for	1,279,430	0.9999036

Figure 12 shows some peculiar examples of successful application of our system. Again, on the left side of the figure, original BNC sentences are presented. All these sentences are correct from a grammatical standpoint, and then no atypical subsequences have been detected. Hence, all cases on the left are true negatives. On the right side of the figure, the same sentence of the left side is presented with some grammatical mistake. However, such grammatical mistakes are not considered atypical by the system, in terms of usage. Moreover, it has been verified that, for a given mistake, in all cases found by the system the subsequences with the grammatical mistake have been employed with the same meaning as in the original sentence. Hence, all cases on the right are true negatives.

Table 4 shows some values related to the sentences of Figure 12c. Here, it can be easily noticed that the subsequence “for an year” (represented in boldface) is characterized by a weight value $w_{k,h}$ higher than the corresponding scaling factor *f*, and then it is considered as a typical subsequence.

Table 4. Values related to the sentences of Figure 12c.

The price of the bow was as much as the income of a common man for a year.		
n. of subsequences: 7		
n. of clusters: 2		
Xie-Beni index: 0.00015013		
f = 47,763		
n-grams	$\|G\|_{\Theta,\pi}$	$w_{k,h}$
the price of	2,175,898,647	0.9999531
of the bow was	3,178,377	0.9999937
was as much	235,902,800	0.9999546
much as the income	223,002,490	0.9999547
income of a	46,863,789	0.9999612
a common man for	784,401	0.9989659
for a year	1,944,067,782	0.9999531

The price of the bow was as much as the income of a common man for an year.		
n. of subsequences: 7		
n. of clusters: 2		
Xie-Beni index: 0.00047737		
f = 46,648		
n-grams	$\|G\|_{\Theta,\pi}$	$w_{k,h}$
the price of	2,175,799,647	0.9996993
of the bow was	643,519	0.9990948
was as much	85,217,200	0.9997106
much as the income	65,702,378	0.9997141
income of a	5,725,105	0.9998504
a common man for	495,611	0.9979714
for an year	**3,010,079**	**0.9999428**

Thus far, we have shown true positive and true negative cases. Figure 13 shows some examples of unsuccessful application of our system. Again, on the left side of the Figure, original BNC sentences are presented. All these sentences are correct from a grammatical standpoint, and then no atypical subsequences have been detected. Hence, all cases on the left are true negatives. On the right side of the figure, the same sentence of the left side is presented with some grammatical mistake. However, such grammatical mistakes are not considered atypical by the system, in terms of usages. Moreover, it has been discovered that in the most cases found by the system, the subsequences with the grammatical mistakes were employed with a different meaning with respect to the original sentence. Hence, all cases on the right are false positives.

For example, some sentences with a different meaning with respect to the sentences of Figure 13a-d are: (a) "*one of* those was one"; (b) "*the opinions expressed in* it do not reflect"; (c) "*the opinion of you* does not reflect"; (d) "*if the* whole thing were". To solve this kind of problems, other constraints can be included in the search. For instance, when rating an initial/final subsequence of a sentence, only initial/final subsequences in the precise usage should be considered valid. For this reason, as a future works we will improve the precise usage calculation with additive features, so as to allow a more exact matching of the meaning.

From the above examples, it becomes then obvious that the test of the performance of our emergent system for text analysis cannot be carried out by means of automatic tools. Indeed, there are no cognitivist models of the observations available, and then the effectiveness of the system must be currently based on human observers.

We have measured the system performance by considering 80 sentences derived from the BNC as described at the beginning of this Section. As metrics, we adopted *Precision* (*P*), *Recall* (*R*), and *F-measure* (*F*) [4], defined as follows:

$$P = \frac{number\ of\ \ correct\ suggestions\ returned}{number\ of\ \ suggestions\ returned} = \frac{true\ positive}{true\ positive\ +\ false\ positive} \tag{12}$$

$$R = \frac{number\ of\ \ correct\ suggestions\ returned}{total\ number\ of\ \ errors\ in\ the\ collection} = \frac{true\ positive}{true\ positive + false\ negative} \tag{13}$$

$$F = 2\frac{P \cdot R}{P + R} \tag{14}$$

Precision is a measure of exactness or quality, whereas recall is a measure of completeness or quantity. The F-measure combines precision and recall via the harmonic mean of them.

Table 5 and Table 6 show the confusion matrix and the system performance, respectively. Both recall and precision are very high, thus confirming the effectiveness of our method.

Table 5. Confusion matrix.

	Actual class (observation)	
Expected class (expectation)	44 True positive (wrong sentence, atypical subsequence discovered)	3 False positive (good sentence, atypical subsequence discovered)
	6 False negative (wrong sentence, nothing discovered	27 True negative (good sentence, nothing discovered)

Table 6. Performance of the system.

P	R	F
0.94	0.88	0.91

7. Conclusions and Future Works

In this paper we presented a novel approach to text analysis able to overcome the designer-dependent representations of the available analyzers, which are more efficient but work as long as the system does not have to stray too far from the conditions under which these explicit representations were formulated. By using an emergent paradigm, in our approach interactions between words in the Web can be represented in terms of visual properties of the input text. In contrast, both symbolic and statistical approaches are cognitivist, involving a representation of a given pre-determined linguistic objective, established based on domain knowledge acquisition in the design process. Hence, cognitivist approaches are characterized by efficiency in solving specific application problems with more or less adaptability, in contradistinction with the emergent approach, which is characterized by embodiment, adaptation, autonomy, and self-organization.

Our approach to text analysis is based on the principles of connectionism and embodiment with the environment. The system employs hit counts and snippets provided by web search engines, in order to rate the subsequences of the input sentence, thus producing usage relationships between words of the sentence. A connectionist structure is then built to represent and optimize such relationships, via an unsupervised fuzzy clustering process. Finally, a visual output of the sentence is provided, with usage

information. The system has been discussed and tested on a collection of sentences of the British National Corpus, showing its effectiveness in highlighting real-world spelling errors. Work is underway to improve the match between word segments and snippets, and to test the system with other languages. Indeed, the approach is completely grammarless and open-world, thus providing an efficient means of analysis of sublanguages in the Web. Moreover, a more usable and manageable version of the system is under development, to allow performing beta tests and collecting assessments of linguistics experts. Finally, a challenge for the future lies in studying the possibility of integration of our method with other web-based models.

Figure 11. Some examples of successful application of our text analysis to the British National Corpus (BNC) data.

	Original BNC sentence	Altered BNC sentence
(a)	I SUSPECT THEY'RE BETTER AT THIS STAGE THAN THE SAME TIME LAST YEAR	I SUSPECT THEY'RE BETTER AT THESE STAGE THAN THE SAME TIME LAST YEAR
(b)	IT WAS THE COLDEST TIME OF THE YEAR AND THE FIRST THING I INSISTED ON WAS HAVING A TELEPHONE	IT WAS THE COLDEST TIME OF THE YEAR AND THE FIRST THING I INSISTED ON WAS HAVES A TELEPHONE
(c)	THAT WAS ONE MAN HE WANTED PEOPLE TO GRIEVE FOR	THAT WAS ONE MAN HE WANTED PEOPLE TO GRIEVING FOR
(d)	SUFFICE TO SAY THAT TIM WON A LOT OF RESPECT FROM A LOT OF PEOPLE THAT DAY	SUFFICE TO SAID THAT TIM WON A LOT OF RESPECT FROM A LOT OF PEOPLE THAT DAY
(e)	IT REALLY IS, THIS IS THE THING THAT PEOPLE WON'T BELIEVE	IT REALLY IS, THIS IS THE THING THAT PEOPLE WON'T BELIEVES
(f)	WHEN I WAS A CHILD, FEW PEOPLE HAD CARS AND BUSES WERE INFREQUENT	WHEN I WAS A CHILD, FEW PEOPLE HAS CARS AND BUSES WERE INFREQUENT
(g)	THE SKY WAS TURNING SEVERAL SHADES OF BLUE AS NIGHT GAVE WAY TO DAY	THE SKY HAVE TURNING SEVERAL SHADES OF BLUE AS NIGHT GAVE WAY TO DAY

(h)	A MAN KILLS THE THING HE LOVES, AND HE MUST DIE A LITTLE HIMSELF	A MAN KILLS THE THING HE LOVES, AND HE MUST DIES A LITTLE HIMSELF
(i)	THE MAN AND THE CHILD ON THE OTHER SIDE WATCHED HIM WITH INTEREST	THE MAN AND THE CHILD ON THE OTHER SIDE WATCHED HE WITH INTEREST
(j)	I WANT TO TOUCH YOU THE WAY NO MAN HAS EVER TOUCHED YOU	I WANT TO TOUCH YOU THE WAY NO MAN HAS EVER TOUCHING YOU
(k)	MY LITTLE CAT, AS YOU SEE, LIKES TO PLAY WITH MY DOG	MY LITTLE CATS, AS YOU SEE, LIKES TO PLAY WITH MY DOG
(l)	MY MOTHER KNOWS THAT MY FAVOURITE FOODS, AND YOU SURELY AGREE WITH HER, ARE PIZZA AND CHOCOLATE	MY MOTHER KNOWS THAT MY FAVOURITE FOODS, AND YOU SURELY AGREE WITH HER, IS PIZZA AND CHOCOLATE

Figure 12. Some peculiar examples of successful application of our text analysis to the BNC data.

	Original BNC sentence	**Altered BNC sentence**
(a)	I THINK IT IS TIME THE THING WAS PUT A STOP TO	I THINK IT ARE TIME THE THING WAS PUT A STOP TO
(b)	THEY RAN OUT TWO THIRDS OF THE WAY THROUGH THE YEAR	THEY RUNS OUT TWO THIRDS OF THE WAY THROUGH THE YEAR
(c)	THE PRICE OF THE BOW WAS AS MUCH AS THE INCOME OF A COMMON MAN FOR A YEAR	THE PRICE OF THE BOW WAS AS MUCH AS THE INCOME OF A COMMON MAN FOR AN YEAR
(d)	NOW MILLIONS OF VISITORS COME EVERY YEAR FOR DAY TRIPS AND HOLIDAYS	NOW MILLION OF VISITORS COME EVERY YEAR FOR DAY TRIPS AND HOLIDAYS

(e)	FOR THE FOLLOWING YEAR, THE SAME THING WILL HAPPEN	FOR THE FOLLOWING YEAR, THE SAME THING WILL HAPPENS
(f)	SHE SAID ONE OF THEM SPOKE TO HER OWN FOUR YEAR OLD CHILD	SHE SAID ONE OF THEY SPOKE TO HER OWN FOUR YEAR OLD CHILD
(g)	THAT WAS ONE MAN HE WANTED PEOPLE TO GRIEVE FOR	THAT WAS ONE MEN HE WANTED PEOPLE TO GRIEVE FOR
(h)	SUFFICE TO SAY THAT TIM WON A LOT OF RESPECT FROM A LOT OF PEOPLE THAT DAY	SUFFICE TO SAY THAT TIM WON A LOT OF RESPECT FROM A LOT OF PEOPLE THOSE DAY
(i)	I WANT TO TOUCH YOU THE WAY NO MAN HAS EVER TOUCHED YOU	I WANTS TO TOUCH YOU THE WAY NO MAN HAS EVER TOUCHED YOU
(j)	THE SKY WAS TURNING SEVERAL SHADES OF BLUE AS NIGHT GAVE WAY TO DAY	THE SKY WAS TURN SEVERAL SHADES OF BLUE AS NIGHT GAVE WAY TO DAY
(k)	AND ON THIS THIRD DAY THE WHOLE THING WAS STILL A BLUR	AND ON THESE THIRD DAY THE WHOLE THING WAS STILL A BLUR

Figure 13. Some examples of unsuccessful application of our text analysis to the BNC data.

	Original BNC sentence	**Altered BNC sentence**
(a)	THAT WAS ONE MAN HE WANTED PEOPLE TO GRIEVE FOR	THOSE WAS ONE MAN HE WANTED PEOPLE TO GRIEVE FOR

(b)	IT DOES NOT REFLECT THE WAY THIS WHOLE THING HAPPENED	IT DO NOT REFLECT THE WAY THIS WHOLE THING HAPPENED
(c)	IT DOES NOT REFLECT THE WAY THIS WHOLE THING HAPPENED	YOU DOES NOT REFLECT THE WAY THIS WHOLE THING HAPPENED
(d)	AND ON THIS THIRD DAY THE WHOLE THING WAS STILL A BLUR	AND ON THIS THIRD DAY THE WHOLE THING WERE STILL A BLUR

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Kilgarriff, A.; Grefenstette, G. Introduction to the Special Issue on the Web as Corpus. *Comput. Linguist.* **2003**, *29*, 333–348.
2. Chesñevar, C.I.; Sabat é-Carrov é, M.; Maguitman, A.G. An argument-based decision support system for assessing natural language usage on the basis of the web corpus. *Int. J. Intell. Syst.* **2006**, *21*, 1151–1180.
3. Hashimoto, T. Usage-based Structuralization of Relationships between Words. In Proceedings of the Fourth European Conference on Artificial Life (ECAL), Brighton, UK, 28–31 July, 1997; The MIT Press: Cambridge, MA, USA, 1997; pp. 483–492.
4. Ellis, N. Constructions, chunking, connectionism: The emergence of second language structure. In *The handbook of second language acquisition*; Doughty, C.J., Long, M.H., Eds.; Blackwell: Malden, MA, USA, 2003; pp. 63–103.
5. Ellis, N. Emergentism, connectionism and language learning. *Lang. Learn.* **1998**, *48*, 631–664.
6. Hopper, P.J. Emergent grammar. In *The new psychology of language: Cognitive and functional approaches to language structure*; Tomasello, M., Eds.; Lawrence Erlbaum: Mahwah, NJ, USA, 1998; pp. 155–175.
7. Liu, V.; Curran, J.R. Web Text Corpus for Natural Language Processing. In Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy, 3–7 April; ACL Anthology; Stroudsburg, PA, USA, 2006; pp. 233–240.
8. Baroni, M.; Bernardini, S.; Ferraresi, A.; Zanchetta, E. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Lang. Resour. Eval.* **2009**, *43*, 209–226.
9. Vernon, D.; Metta, G.; Sandini, G. A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. *IEEE T. Evolut. Comput.* **2007**, *11*, 151–180.

10. Liddy, E. Natural language processing. In *Encyclopedia of Library and Information Science*, 2nd ed.; Marcel Dekker: New York, NY, USA, 2003; pp. 2126–2136.
11. Shaalan, K. Rule-based Approach in Arabic Natural Language Processing. *The Int. J. Inform. Comm. Tech.* **2010**, *3*, 11–19.
12. Helbig, H. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, Germany, 2006; Volume XIX, pp. 395–433.
13. Carlson, A.; Mitchell, T.M.; Fette, I. *Data analysis project: Leveraging massive textual corpora using n-gram statistics*. Technical Report CMU-ML-08–107, Machine Learning Department; Carnegie Mellon University: Pittsburgh, PA, USA, 2008.
14. Islam, A.; Inkpen, D. Real-word spelling correction using Google Web 1T *n*-gram with backoff. In Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), Dalian, China, 24–27 September, 2009; IEEE Computer Society: Piscataway, NJ, USA, 2009; pp.1–8.
15. Islam, A.; Inkpen, D. Near-Synonym Choice using a 5-gram Language Model. *Comput. Sci.* **2010**, *46*, 41–52.
16. Arppe, A.; Gilquin, G.; Glynn, D.; Hilpert, M.; Zeschel, A. Cognitive Corpus Linguistics: five points of debate on current theory and methodology. *Corpora* **2010**, *5*, 1–27.
17. Taira, R.K.; Soderland, S. A statistical NLP for medical reports. In Proceedings of the American Medical Informatics Association Fall Symposium (AMIA), Washington, DC, USA, 6–10 November, 1999; Hanley & Belfus Inc.: Philadelphia, PA, USA, 1999; pp. 970–974.
18. Gilquin, G.; Gries, S.T. Corpora and experimental methods: a state-of-the-art review. *Corpus Linguist. Linguist. Theory* **2009**, *5*, 1–26.
19. Inkpen, D.; Islam, A. Unsupervised Approaches to Text Correction using Google *n*-grams for English and Romanian. In *Multilinguality and Interoperability in Language Processing with Emphasis on Romanian*; Tufis, D., Forascu, C., Eds.; Romanian Academy Publishing House: Bucharest, Romania, 2010; pp. 270–285.
20. Johnson, M. How the statistical revolution changes (computational) linguistics. In Proceedings of the Workshop on the Interaction between Linguistics and Computational Linguistics (EACL-ILCL), Athens, Greece, 30–31 March, 2009; Association for Computational Linguistics: Stroudsburg, PA, USA, 2009; pp. 3–11.
21. Jacquemont, S.; Jacquenet, F.; Sebban, M. Correct your text with Google. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI), Silicon Valley, USA, 2–5 November, 2007; IEEE Press: Piscataway, NJ, USA, 2007; pp. 170–176.
22. Rieger, B.B. Meaning acquisition by semiotic cognitive information processing systems. In Proceedings of the 3rd International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society (ISUMA-NAFIPS), College Park, MD, USA, 17–20 September, 1995; IEEE Computer Society: Los Alamitos, CA, USA, 2005; pp. 390–395.
23. Tromp, E.; Pechenizkiy, M. Graph based *n*-gram language identification on short texts. In Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands (Benelarn), The Hague, Netherlands, 20 May, 2011; Leiden University and the Kecida Group; The Hague, Netherlands, 2011; pp. 27–34.

24. Thomas, M.S.C.; McClelland, J.L. Connectionist models of cognition. *Cognitive Modeling Paradigms*. Cambridge handbook of computational cognitive modeling; Sun, R., Ed.; Cambridge University Press: Cambridge, England, UK, 2008; pp. 23–58.
25. Lee, B.; Riche, N.H.; Karlson, A.K.; Carpendale, S. SparkClouds: Visualizing Trends in Tag Clouds. *IEEE T. Vis. Comput. Gr.* **2010**, *16*, 1182–1189.
26. Palmer, M.; Gildea, D.; Kingsbury, P. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Comput. Linguist.* **2005**, *31*, 71–106.
27. Lapata, M.; Keller, F. Web-based models for natural language processing. *ACM T. Speech Lang. Proc.* **2005**, *2*, 1–31.
28. Bergsma, S.; Lin, D.; Goebel, R. Web-scale *n*-gram models for lexical disambiguation. In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI), Pasadena, CA, USA, 11–17 July, 2009; AAAI Press: Danvers, MA, USA, 2009; pp. 1507–1512.
29. Izumi, E.; Uchimoto, K.; Saiga, T. Automatic error detection in the Japanese learners' English spoken data. In the Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics (ACL): Sapporo, Japan, 7–12 July, 2003; ACL, Stroudsburg, PA, USA, 2003; pp. 145–148.
30. Hirst, G.; Budanitsky, A. Correcting real-word spelling errors by restoring lexical cohesion. *Nat. Lang. Eng.* **2005**, *11*, 87–111.
31. Schaback, J.; Li, F. Multi-level feature extraction for spelling correction. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, 6–12 January, 2007; Morgan Kaufmann Publishers: Burlington, MA, USA, 2007; pp. 79–86.
32. Wilcox-O'Hearn, L.A.; Hirst, G.; Budanitsky, A. Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model. In Proceedings of 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Haifa, Israel, 17–23 February, 2008; Springer-Verlag: Berlin Heidelberg, Germany, 2008; Lecture Notes in Computer Science; Volume 4919, pp. 605–616.
33. Hodge, V.J.; Austin, J. A comparison of standard spell checking algorithms and a novel binary neural approach. *IEEE Knowl. Data Eng.* **2003**, *15*, 1073–1081.
34. Mundada, P.; Ghotkar, A. An approach to second generation tag cloud for assessment of business search. In Proceedings of the IEEE International Conference on Technology Enhanced Education (ICTEE), Kerala, India, 3–5 January 2012; IEEE Press: Piscataway, NJ, USA 2012; pp.1–3.
35. Miyamoto, S.; Ichihashi, H.; Honda, K. Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications. In *Studies in Fuzziness and Soft Computing*; Springer-Verlag: Berlin Heidelberg, Germany, 2008.
36. Kruse, R.; Döring, C.; Lesot, M.-J. Fundamentals of Fuzzy Clustering. In *Advances in Fuzzy Clustering and its Applications*; de Oliveira, J.V., Pedrycz, W., Eds.; John Wiley & Sons: Chichester, UK, 2007.
37. Cimino, M.G.C.A.; Lazzerini, B.; Marcelloni, F. A novel approach to fuzzy clustering based on a dissimilarity relation extracted from data using a TS system. *Pattern Recognit.* **2006**, *39*, 2077–2091.
38. Kim, D.-W.; Lee, K.H.; Lee, D. On cluster validity index for estimation of the optimal number of fuzzy clusters. *Pattern Recognit.* **2004**, *37*, 2009–2025.

39. Sun, H.; Wang, S.; Jiang, Q. FCM-Based Model Selection Algorithms for Determining the Number of Clusters. *Pattern Recogn.* **2004**, *37*, 2027–2037.
40. Burnard, L.; Aston, G. *The BNC Handbook: Exploring the British National Corpus with SARA*; Cambridge University Press: Cambridge, UK, 1998. Available online: http://www.natcorp.ox.ac.uk (accessed on 12 September 2013).
41. Leech, G.; Rayson, P.; Wilson, A. *Word Frequencies in Written and Spoken English: based on the British National Corpus*; Longman: London, UK, 2001. Available online: http://ucrel.lancs.ac.uk/bncfreq (accessed on 12 September 2013).