Mehler, Alexander (2002):
Hierarchical Analysis of Text Similarity Data.
In: Künstliche Intelligenz (KI) 2, 12-16.

Hierarchical Analysis of Text Similarity Data

Alexander Mehler

Abstract

Semantic spaces are used as a representational format for modeling similarities of signs. As a multidimensional data structure they are bound to the question of how to explore similarity relations of signs mapped onto them. This paper introduces an abstract data structure called *dependency scheme* as a *formal format* which encapsulates two types of order relations, whose variable instantiation allows to derive different *classes* of trees for the hierarchical analysis of text similarity data derived from semantic spaces.

1 Introduction

Semantic spaces have been proposed as a highdimensional numerical format for representing semantic similarities of signs. They prove to be efficient in computational linguistics [9, 11], computational psychology [4, 5], and IR [3]. These approaches have in common that they use numerical measures for modeling sign similarities: *words* are judged to be similar to the degree that they occur in similar contexts, whereas *texts* are judged to be similar to the extent that they have semantically similar constituents. A touchstone for the effectiveness of semantic spaces are indirect meaning relations: they allow to interrelate words even if they never co-occur, but tend to occur in similar contexts. Furthermore, texts are judged to be similar even if they only share function words, but deal with similar contents.

A fundamental question raised in this context is how to explore similarities of signs mapped onto semantic spaces. In case that texts are represented as points in semantic space, this question forms a text mining task called *text linkage* [8], which refers to the exploration of implicit, content based relations of texts and their annotation as typed links in hypertexts. In linguistic terms, these relations are *intertextual* in the sense that they are neither necessarily explicit, nor do they necessarily underly text production, but may support text reception by embedding texts into textual contexts. In order to narrow down a mining algorithm for solving the task of text linkage, three criteria are referred to:

1. *Thematic progression:* Intertextual relations go beyond pairwise text linkage by ordering texts as manifestations of thematic progression; e.g. sequences of chronologically ordered news dealing with the same topic. Thus, the text linkage algorithm needs to produce correlates of these higher level structures.

2. *Linguistic interpretability:* Besides thematic progression, the concept of *priming* guides linkage: in contrast to word priming, where *single* primes are used to *associate* related signs, *context* priming refers to the fact that textual contexts of text components sustain appropriate, while suppressing inappropriate senses [12]. Thus, the linkage algorithm needs to produce text sequences in which preceding texts prime succeeding ones.

3. *Contextsensitivity:* According to the dynamics of human information processing, a text corpus does not have a fixed hypertext structure. Thus, a flexible information structure is needed which allows to dynamically link texts dependent on varying contexts.

In literature many techniques for visualizing highdimensional feature spaces are discussed: representing a sign's environment by means of *lists* runs the risk of successively ordering thematically diverse units. Obviously, lists neglect the poly-hierarchical structure of semantic spaces which may induce divergent thematic progressions starting from the same polysemous unit. Salton et al. [10] describe a method called *breadth m-depth n*-search for generating *trees*: starting from a focal unit, its m nearest neighbors in space are determined as its immediate successors. This procedure is repeated for all successors until the tree's height equals n. A central problem of this method relates to the question of how to specify and linguistically interpret the parameters m and n. A data structure which overcomes this deficiency is given by *minimal spanning trees* (MST) used in the area of hypertext authoring [2]. As will be shown, a central problem of MSTs relates to the purely associative links they produce. An alternative is given by *networks* also used in hypertext area [1]. Networks pose the problem that they rapidly get complex even for small sets of units. Finally, *cluster analysis* may be used. Above all, they face the problem of how to name the resulting clusters.

This paper proposes *cohesion trees* (CT) as a hierarchical data structure for exploring semantic spaces. In contrast to the latter techniques, their central organization units are chains of texts: suppose a text x to be inserted into a tree T visualizing dependencies in semantic space and two paths $P_1 = (a_1, \ldots, a_n)$, $P_2 = (b_1, \ldots, b_m)$ of textual nodes of T to which x may be attached. Suppose now a measure ξ evaluating the cohesion of paths P_i based on the meaning representations of their constituents in the space. Finally suppose that ξ states that (because of indirect relations of texts a_i, b_j, $i < n$, $j < m$, and x) path P_2x is more cohesive than P_1x, although the pair (a_n, x) is more cohesive than (b_m, x). In order to augment cohesion not only of text pairs, but of text chains, the CT generates path P_2x instead of P_1x. Thus, CTs do not only reflect dependencies of directly linked, but cohesion relations of indirectly linked nodes. A traversal through a semantic space produced by a CT is bound to the requirement that it is cohesive, i.e. thematically homogeneous to a descending degree as the path grows. In this sense cohesion serves as a linguistic criterion for hierarchical text linkage: the more similar the lexical organization of two texts, the more probably they are linked *provided that the path into which they enter progresses thematically, if they are attached to it.* CTs shift the perspective from binary text links to whole paths of such links. They are formally described as instances of the so called *dependency scheme*, whose application area are two-level-hypertext systems for the poly-hierarchical traversal of the same document collection starting from varying focal texts.

2 The Dependency Scheme

As a formalism for hierarchical data analysis, the *dependency scheme* is defined as an abstract class of trees encapsulating two types of order relations for modeling priming effects. The variable instantiation of these relations allows the derivation of different types of tree-like interpretation structures reflecting the criterion of thematic progression with varying strength. The scheme is introduced in two steps:

Definition 1 Let V be a set of signs represented as feature vectors $\vec{x}_i \in X \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$, and $\delta : X^2 \rightarrow \mathbb{R}_0^+$ a symmetric distance measure—vectors assigned to $x_i \in V$ are always referred to via $\vec{x}_i$. E.g. X is a set of text vectors in Salton's model [10], V a text corpus, and δ a measure derived from the cosine coefficient. The *complete weighted undirected graph* $G = \langle V, E, \omega \rangle$ induced by (X, δ) is a graph with edge set $E = \{\{v, w\} \mid v \neq w\} \subseteq V^2$, $|E| = \frac{n(n-1)}{2}$, and weighting function $\omega(\{v, w\}) = 1 - \delta(\vec{v}, \vec{w})/\mathrm{M}(\delta)$. $\mathrm{M}(\delta)$ is the maximal value δ assumes.

Definition (1) does not demand that δ is a metric. It only assumes that there is a distance measure for comparing signs in V. Moreover, δ does not need to be a total function. If (X, δ) only induces a connected graph, a complete weighted graph can be derived by introducing edges between unconnected nodes whose weight is a function of the shortest path between these nodes in the input graph. Thus, the semantic space models listed above induce weighted graphs $G = \langle V, E, \omega \rangle$ which can be used for the derivation of $2^{|V^2|/2}$ different undirected graphs. On this background the question arises, whether there exists a subset of "meaningful" trees in the sense that their construction can be interpreted in linguistic terms. Instead of using *informal* terms, this subset is *formally* approximated by means of the *dependency scheme* in order to guarantee unambiguity and correctness of linguistic interpretation as well as extendability of formalization:

Definition 2 Let $G = \langle V, E, \omega \rangle$ be a graph induced by (X, δ) according to def. (1), $x \in V$ a node, and $\Omega = \langle \leq_x^1, \{\leq_y^2 \mid y \in V \setminus \{x\}\} \rangle$ a tuple, where $\leq_x^1, \leq_y^2 \subset V^2$ are linear order relations with infimum x and $y \in V \setminus \{x\}$, respectively. The graph $\mathcal{D}(G, x, \Omega) = \langle V, \mathcal{E}, \nu \rangle$ with $\mathcal{E} = \{\{v, w\} \mid v <_x^1 w \wedge \neg\exists y \in V : y <_x^1 w \wedge y <_w^2 v\}$ and $\nu : \mathcal{E} \rightarrow \mathbb{R}$ (the restriction of ω to E) is called Ω-*tree induced by* x. For variable G, x and Ω, $\mathbf{D}(G, x, \Omega)$ is called *dependency scheme*. $v <_y^i w$, where $i \in \{1, 2\}$, $v, w, y \in V$, abbreviates $v \neq w \wedge v \leq_y^i w$.

Theorem. *For a complete weighted undirected graph* $G = \langle V, E, \omega \rangle$, *node* $x \in V$, *and* $\Omega = \langle \leq_x^1, \{\leq_y^2 \mid y \in V \setminus \{x\}\} \rangle$, *the instance* $\mathcal{D}(G, x, \Omega)$ *of the dependency scheme is a tree.*

The proof of this theorem is left because of lack of space. The scheme is based on two types of order relations: (i) Relations of type $\leq_x^1$ model priming effects induced by root x of $\mathcal{D}(G, x, \Omega)$. $\leq_x^1$ determines the order, in which nodes $y \in V$ are inserted into the tree: the more similar x and y, the shorter the distance $\delta(x, y)$, the shorter the path between x and y in the tree. In other words: units "primed" by x because of higher similarity are inserted "earlier". (ii) Relations of type $\leq_y^2$ model priming effects induced by a node's predecessor: y is attached to the infimum $\inf_{\leq_y^2}(V_i)$ of the set V_i of nodes already inserted into the tree. The variable specification of $\leq_y^2$ allows the derivation of tree-like structures, which diverge with respect to the type of priming they model as shown in the following.

2.1 Dependency Trees

Dependency trees serve for the perspective visualization of semantic spaces using varying roots as temporary, local centers. They are introduced by instantiating the kernel order relations of the dependency scheme by means of an alphabetic order $\leq_x^a$ with infimum x:

Definition 3 Let $G = \langle V, E, \omega \rangle$ be a graph induced by (X, δ) according to def. (1), and x a node. $\leq_x \subset V^2$ is a linear order, which orders nodes with respect to the distances of their representations in X: $v \leq_x w$ iff $\delta(\vec{v}, \vec{x}) < \delta(\vec{w}, \vec{x}) \vee (\delta(\vec{v}, \vec{x}) = \delta(\vec{w}, \vec{x}) \wedge v \leq_x^a w)$.

$v \leq_x w$ means that the pair of signs v, x is more similar than w, x. In case of a semantic space (X, δ) underlying G, $v \leq_x w$ says that the meaning representation $\vec{v}$ of v is closer to $\vec{x}$ than the meaning representation $\vec{w}$ of w. In other words: compared with $\{x, w\}$, $\{x, v\}$ realize more similar usage regularities. x is the infimum of $\leq_x$, which is reflexive, antisymmetric, transitive, and linear. Hence, $\leq_x$ defines a chain over V. $\leq_x$ orders signs relative to x, whose meaning representation serves as a *local center* of the semantic space under consideration.

Definition 4 Let $G = \langle V, E, \omega \rangle$ be a graph according to definition (1) and $x \in V$ a node. For $\Omega = \langle \leq_x, \{\leq_y | \, y \in V \setminus \{x\}\}\rangle$, the Ω-tree $D(G, x, \Omega)$ is called *dependency tree* (DT).

In case where the underlying space (X, δ) is a semantic space, DTs represent the environment of a sign's meaning representation as a tree. DTs have been algorithmically invented by Rieger [9], whereas Lin [6] proposes an informal definition of DTs. Both approaches restrict the set of vertices to the set of words and lack algebraic specifications of DTs. Furthermore, their connection with other tree structures are left out as well as their systematic evaluation. The basic mechanism underlying the organization of DTs is *context free association* together with an initial ordering under the control of x's (i.e. the chosen root's) meaning representation: (1) The order of signs to be inserted into the dependency tree $D(G, x, \Omega)$ is determined by $\leq_x$ and thus by the distances of the nodes' meaning representations with respect to $\vec{x}$. (2) Any sign $w \in V$ is linked with that sign v already inserted into the tree which is not only closer to x than w (i.e. $v \leq_x w$), but also closer to w than any other vertex y with $y \leq_x w$. In other words, v is connected with its strongest associate in the sense of the (dis-)similarity data modeled by δ already inserted into the tree.

2.2 Cohesion Trees

If the vertex set is restricted to lexical units, DTs can be said to model word priming effects, where the root serves as an initial prime: DTs result from a process of spreading activation *without* taking path context into account. This deficiency, which in case of textual vertices may cause the generation of incohesive, thematically diverging chains of texts, is overcome with the help of the concept of *cohesion tree* as a further instance of the dependency scheme. As in case of DTs, CTs are introduced by instantiating the dependency scheme's constitutive order relations. This is done with the help of the concept of *path sensitive distance*:

Definition 5 Let $\langle V, E \rangle$ be a graph. A sequence $P_{v_1 v_n} = (v_1, \ldots, v_n)$ of vertices $V(P_{v_1 v_n}) = \{v_1, \ldots, v_n\} \subseteq V$ is called *path* from v_1 to v_n, if for each pair of vertices $v_i, v_{i+1} \in V(P_{v_1 v_n})$ there exists a distinct edge $\{v_i, v_{i+1}\} \in E$. v_1 is the *start* and v_n the *end* vertex of $P_{v_1 v_n}$. All other vertices are *inner*. $P_{v_1 v_n}$ is *cyclic*, if $v_1 = v_n$. P is *simple*, if all its inner vertices are distinct.

Definition 6 Let $G = \langle V, E, \omega \rangle$ be a graph induced by (X, δ) according to definition (1) and $P = (v_1, \ldots, v_k)$ a simple path in G. The *path sensitive distance* $\delta^*(P, x)$ of $x \in V$ with respect to P is defined as $\delta^*(P, x) = \frac{1}{\mathrm{M}(\delta)} \sum_{v_i \in V(P)} \omega_i \delta(\vec{x}, \vec{v_i}) \in [0, 1]$, where $\sum_{v_i \in V(P)} \omega_i \leq 1$. $\mathrm{M}(\delta)$ is the maximal value δ assumes.

Definition 7 Let $G = \langle V, E, \omega \rangle$ be a graph induced by (X, δ) according to definition (1) and $\mathcal{P}$ the set of all simple paths in G with start vertex x and $y \in V$ a node. The well-ordering $\sqsubseteq_y \subseteq \mathcal{P}^2$ orders all paths in $\mathcal{P}$ with respect to y: $P_{xv} \sqsubseteq_y P_{xw}$ iff $\delta^*(P_{xv}, y) < \delta^*(P_{xw}, y) \vee (\delta^*(P_{xv}, y) = \delta^*(P_{xw}, y) \wedge v \leq_y w)$.

Definition 8 Let $G = \langle V, E, \omega \rangle$ be a graph according to definition (1) and $x \in V$ a node. For $\Omega = \langle \leq_x, \{\sqsubseteq_y \mid y \in V \setminus \{x\}\}\rangle$, the Ω-tree $C(G, x, \Omega)$ is called *cohesion tree* (CT).

In def. (6) the impact of indirectly linked nodes are modeled with the help of bias ω_i as an implicit parameter of def. (8). Computing ω_i includes, but is not limited to 3 alternatives:

1. *Order independence:* if ω_i is constant for all $v_i \in V(P)$, then the position of a vertex in path P is seen to be irrelevant.

2. *Context sensitivity:* if ω_i is increasing with path length, the syntagmatic order of P is reflected in the sense that the shorter the distance of vertex x to a vertex in P, the higher the impact of their distance measured by δ. In other words: the descending impact of more distant units allows a weak topic change as the path grows. A function meeting this condition looks as follows: let $P = (v_1, \ldots, v_k)$ be a path of k nodes and $c \in (0, 1]$ a constant, then ω_i, $i = 1, \ldots, k$, is computed as $\omega_i = \frac{1}{\sum_i c^{k-i+1}} c^{k-i+1} \in [0, 1]$.

3. *Context independence:* if all ω_i are set to 0, except w_k, then $C(G, x, \Omega)$ is reduced to a DT. Thus, DTs form a special case of CTs in the sense that they *neglect* path contexts.

Relations of type $\sqsubseteq_y$ model *context priming effects* by specifying the predecessor of a node y to be the end vertex of the most cohesive path P of the set of candidates, to which y may be attached. y is not simply the strongest associate of P's end vertex, but primed by P as a whole. $\sqsubseteq_y$ is based on δ^* which extends δ in the sense that it does not only reflect distances of meaning representations of *pairs*, but of *sequences* of signs. δ^* takes the *syntagmatic order* of paths into account: the more distant two signs in a path, the less their mutual impact, the less their contribution to δ^*. On the other hand, this negative distance effect may be compensated by higher "similarity" of signs, i.e. by lower values of δ. Preceding signs constitute a context which may superimpose or compensate the impact of a path's end vertex. This context effect retrogrades as the distance of nodes in the path grows, and consequently, topic changes latently controlled by preceding nodes can be realized.

To summarize: *In contrast to lists*, CTs represent the environment of a sign mapped onto a semantic space as a tree, whose branches are evaluated with respect to their thematic cohesion. *In contrast to MSTs*, CTs do not only reflect direct node-to-node dependencies, but also relations of indirectly linked units. Finally, *in contrast to DTs*, CTs do not only reflect priming induced by single nodes, but context priming induced by whole paths.

2.3 A Sample Analysis

In this section, DTs and CTs are exemplified. It is shown that minimal spanning trees (MST), DTs, and CTs represent *different* concepts. Let $G = \langle \{a, b, c, d\}, E, \omega \rangle$ be a complete weighted undirected graph induced by (X, δ) and δ be a metric over $\{a, b, c, d\}$, where $\delta(c, d) < \delta(b, d) < \delta(b, c) < \delta(a, b) < \delta(a, c) < \delta(a, d)$. This situation is visualized in figure (1, left), where smaller values of δ are represented by shorter distances of nodes. The MST $M(G)$ of G is a tree with edge set $\{\{a, b\}, \{b, d\}, \{d, c\}\}$. In order to compare MSTs and DTs, a dependency tree $\mathcal{D}(G, a, \Omega_a)$, $\Omega_a = \langle \leq_a, \{\leq_b, \leq_c, \leq_d\}\rangle$, is built. First, $T_1 = \langle \{a, b\}, \{\{a, b\}\}\rangle$ is constructed, since $b = \inf_{\leq_a}(\{b, c, d\})$. Two nodes, c and d, are left. Since $\delta(a, c) < \delta(a, d)$ and $b = \inf_{\leq_c}(\{a, b\})$, $T_2 = \langle \{a, b, c\}, \{\{a, b\}, \{\{b, c\}\}\rangle$ is built,

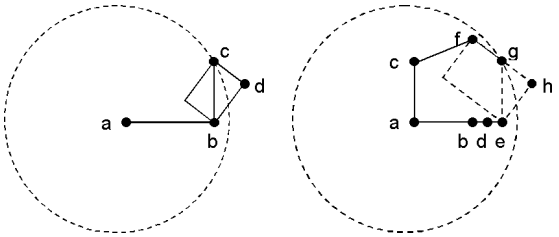

Figure 1: The difference of DTs with respect to MSTs (left), and CTs (right).

which in contrast to $M(G)$ covers the edge $\{b, c\}$. In the 3rd step, $T_3 = \mathcal{D}(G, a, \Omega_a)$ is built extending V_2 by node d and E_2 by edge $\{c, d\}$. Hence, $\mathcal{D}(G, a, \Omega)$ is no MST. In case of graph G it holds that $D(G, a, \Omega_a) \neq D(G, b, \Omega_b) = D(G, c, \Omega_c) = D(G, d, \Omega_d) = M(G)$ for corresponding $\Omega_b, \Omega_c, \Omega_d$. G induces two DTs, but only one MST. In order to compare now DTs and CTs, we suppose a situation as visualized in figure (1, right), where smaller values of δ are represented by shorter edges, whereas labels indicate the order $\leq_a$, in which the nodes are inserted into the DT and CT, respectively. Suppose that a DT $D(G, a, \Omega)$, $\Omega = \langle \leq_a , \{\leq_x | x \in \{b, c, d, e, f, g\}\}\rangle$, and a CT $C(G, a, \Omega')$, $\Omega' = \langle \leq_a, \{\sqsubseteq_x | x \in \{b, c, d, e, f, g\}\}\rangle$, have already been constructed with two paths (a, c, f, g) and (a, b, d, e) each. Now the question arises, to which node h is to be attached. In case of the DT, edge $\{h, g\}$ is constructed, since $\delta(g, h) < \delta(e, h)$. In contrast to this, CTs reflect the path context: distances $\delta(h, e), \delta(h, d)$, $\delta(h, b)$ are compared with $\delta(h, g), \delta(h, f), \delta(h, c)$. If the concrete values of ω_i in definition (6) determine that $e \sqsubseteq_h g$, edge $\{h, e\}$ is finally constructed.

Based on the algorithm for computing semantic spaces described in the next section, a MST and a CT using the same newspaper article as root have been generated (see figure 2). The textual root deals with a *football game* broadcast in so called *pay tv*. In figure (2) each textual node is represented by title and topic category as found in the newspaper. Although both trees start with content related units, the MST rapidly diversifies thematically. Furthermore, no text dealing with the *broadcast* topic is found in the outline of the MST. In contrast to this, the CT comprises two thematically homogeneous branches: one of it comprises texts dealing with *football*, whereas the other covers texts dealing with *football broadcasting rights*. According to this example CTs seem to be more adequate for modeling higher level structures of thematic progression than MSTs. Whether this judgment is persistent has to be shown by systematic evaluation.

3 Evaluation

In order to evaluate the concepts introduced so far, a semantic space of about 2,000 lexical dimensions was computed using a text corpus C of 502 texts (of the `Süddeutsche Zeitung`) following the procedure described in [9] and [8]: (i) Syntagmatic regularities of words $a, b \in W$, $|W| = n$, in C are measured with the help of a correlation coefficient. The correlations are used to map words onto *corpus points* as elements of the so called *corpus space* $\mathcal{C} \subset \mathbb{R}^n$. (ii) Dissimilarities of syntagmatic regularities are mapped by an Euclidian metric δ operating on $\mathcal{C}$. The resulting distance values are used to generate a so called *semantic space* $\mathcal{S} \subset \mathbb{R}^n$. Each word $a_i \in W$ is mapped onto a *meaning point* $\vec{s}_i = (s_{i1}, \ldots, s_{in}) \in \mathcal{S}$ with coordinate values $s_{ij} = 1 - \delta(\vec{c}_i, \vec{c}_j)/(2\sqrt{n}) \in [0, 1]$, where $\vec{c}_i, \vec{c}_j$ are the corpus points of lexemes $a_i, a_j \in W$. (iii)

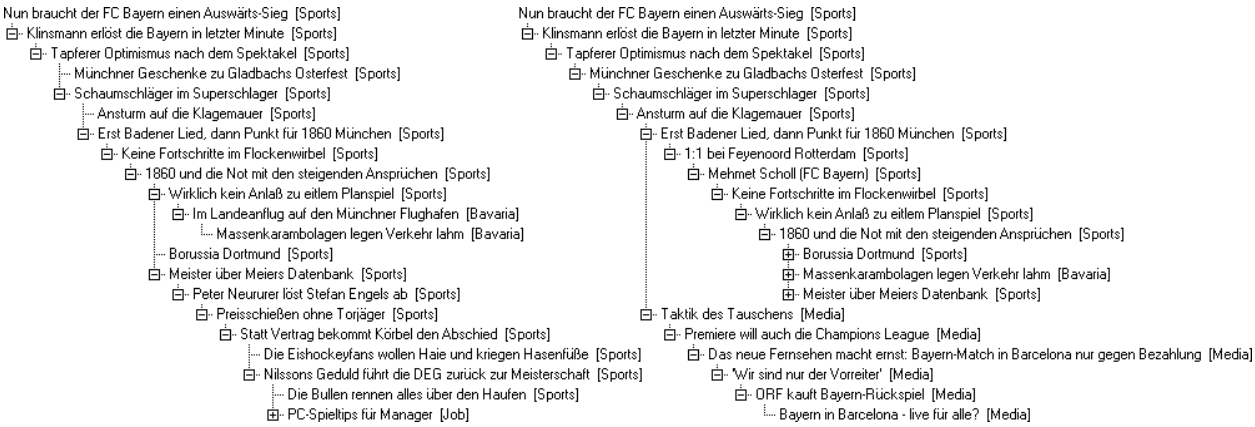

Figure 2: The MST (left) and CT (right) of the text sample.

Finally, texts are mapped onto $\mathcal{S}$ by means of a weighted mean of meaning points assigned to their lexical constituents: $\vec{x}_k = \sum_{a_i \in W(x_k)} w_{ik}\vec{s}_i \in \mathcal{S}$ is the meaning point of text $x_k \in C$, $W(x_k)$ is the set of all types of all tokens in x_k, and w_{ik} is a bias having the same function as the *tfidf*-scores in information retrieval. In linguistic terms, lexical meaning points model paradigmatic usage regularities, whereas textual meaning points model lexical cohesion. As a result of mapping texts onto $\mathcal{S}$, they can be compared regarding the similarity of their lexical organization. This kind of similarity is modeled with the help of a measure σ based on the Euclidean metric δ: $\sigma(\vec{x}, \vec{y}) = 1 - \delta(\vec{x}, \vec{y})/\mathrm{M}(\delta)$. The more similar the paradigmatic usage regularities of the lexical constituents of two texts, the shorter the distance of their meaning points, the more similar the texts.

Next, $(\mathcal{S}, \delta)$ is used to derive a weighted graph $\langle C, E, \omega \rangle$ according to definition (1). For each text $x \in C$ seven types of trees are computed: besides MSTs, DTs, and CTs, so called maximal degree trees (MDT), maximal height trees (MHT), random predecessor trees (RPT), and random successor trees (RST). For vertex set $V = C$, $|V| = m$, the MDT of $x \in C$ is a tree with $m - 1$ leafs, whereas the MHT is a tree of height $m - 1$ with only one leaf, where the nodes are inserted according to the chain induced by $\leq_x$. An RPT using $x \in C$ as its root is a tree, in which nodes $v \in V \setminus \{x\}$ are inserted according $\leq_x$, but where the predecessor of v is randomly chosen under the nodes already inserted. An RST is a tree with root $x \in C$, where the successors and their predecessors are randomly chosen. Whereas MDTs and MHTs model simple list structures of minimal and maximal height, RPTs and RSTs are randomly organized. Both classes of trees are used to evaluate CTs. As a result, 3514 different trees were automatically computed and compared regarding their *cohesion*.

The literature discusses several measures for evaluating weighted trees. The *weighted length*, which is always minimized by MSTs, cannot serve as a measure for the cohesion of trees, since it only considers directly linked nodes and therefore runs the risk to ignore dependencies of indirectly linked units. Furthermore, the *weighted path length* used in the area of binary search trees cannot be applied either, because it only considers leafs and their heights without evaluating inner nodes. In the following, a measure is developed based on three premises: (i) In order to account for cohesion of trees, the measure has to consider dependencies of directly as well as indirectly linked nodes. Thus, *paths* (and not *pairs of nodes* or *leafs* as in case of the weighted length and path length, respectively) are used as the fundamental unit for measuring cohesion: the more similar the nodes of a path—*regardless*

of their order—, the more cohesive the path. (ii) The cohesion of a tree is a function of the cohesion of its paths. (iii) MDTs as well as MHTs assume low cohesion scores: as a result of their structural simplicity, they correspond to simple lists. Premise (i) guarantees that the criterion used for constructing cohesion trees is not used for their evaluative comparison with other trees, since it abstracts from path order. A measure, which takes these constraints into account, looks as follows: Let $D(x) = \langle V, \mathcal{E}, \nu \rangle$ be a tree with root x and $\mathcal{P}$ the set of all paths in $D(x)$ with start vertex x and end vertex $y \in L(D(x))$, where $L(D(x))$ is the set of all leafs of $D(x)$. The cohesion of a path $P_{v_1 v_n} = (v_1, \ldots, v_k) \in \mathcal{P}$ is a function of the average distance of its constituent nodes: the less this distance, the more similar the nodes according to σ, the more cohesive this path. Thus, the cohesion score of path $P_{v_1 v_n}$ is computed as:

$$\xi(P_{v_1 v_k}) = 1 - \lambda \sum_{i=1}^{k} \sum_{j=i+1}^{k} \mathrm{M}(\delta)^{-1} \delta(v_i, v_j) \tag{1}$$

where n is the dimension of the underlying semantic space and $\lambda = \frac{2}{k(k-1)}$ is a scaling factor, which causes that $\xi(P_{v_1 v_k}) \in [0, 1]$. In order to prevent that the MDT (with shortest possible paths) induced by x is assigned the highest cohesion score, the sum is scaled by the number of all distances in the path. Finally, a cohesion value for the whole tree is derived by summing the cohesion scores of all its paths:

$$\kappa(D(x)) = \frac{1}{l} \sum_{P \in \mathcal{P}} \xi(P) \in [0, 1] \tag{2}$$

where $l = |L(D(x))|$ is the number of leafs in $D(x)$. Now, the sum is scaled in order to prevent that the corresponding MHT induced by x is assigned the highest cohesion score.

In table (1), the cohesion scores ($\sum \kappa(D)$) are summarized for each of the tree types and all instances in the test corpus. Average maximum degrees ($\bar{D}$), average (maximum) height ($\bar{H}$), and the sums of weighted lengths $\sum \mathcal{L}(x)$ are shown. The values are interpreted as follows: MSTs are of shortest length, thereby reaching high cohesion scores. DTs are longer, have a comparable average maximum degree, but are much flatter than MSTs. At the same time, they achieve a higher cohesion score. CTs are deeper than DTs, but flatter than MST. They are longer than DTs and have a low average maximum degree. But CTs realize the highest cohesion score. This result is even more obvious if the scores for each text are compared in isolation: in 494 cases, CTs realize the highest score under all tree types. Thus, in the absolute majority of test cases CTs improve the cohesion of their paths compared to all other tree types. Finally, RPTs and RSTs have low cohesion scores, whereas MHTs and MDTs are of lowest cohesion. Thus, even randomly organized trees realize a higher cohesion score than these list-like structures. The case of RPTs shows that even if the predecessor of a node is chosen by chance, to hierarchically organize texts leads to more cohesive paths than to *list* them – the predominant procedure for organizing search results in Internet.

4 Conclusion

A scheme for hierarchical analysis of text similarities and two of its instances were described. Besides DTs, CTs have been introduced as an alternative to lists, cluster analysis and associative MSTs. The scheme is based on order relations, which were linguistically interpreted using the concept of priming. Future work aims at an algebra of tree-like interpretation structures and the replacement of lists as a basis for organizing search results in Internet.

Type	$\bar{D}$	$\bar{H}$	$\sum \mathcal{L}(x)$	$\sum \kappa(D)$	rank
MDT	501	1	9765.7	344.6	7
MHT	1	501	8321.5	344.6	6
RST	28.4	11.4	9796.6	345.2	5
RPT	27.7	10.9	8474.5	385.2	4
MST	8	35.6	3454.0	407.0	3
DT	7.8	22.1	3551.6	410.0	2
CT	5.7	31.5	3678.7	413.8	1

Table 1: Results for the text corpus.

References

[1] Allen, J. 1997. Building Hypertext Using Information Retrieval. *Information Processing & Management*, 33(2): 145-159.

[2] Botafogo, R. A., Rivlin, E. and B. Shneiderman. 1992. Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. *ACM Trans. Inf. Syst.*, 10(2): 142-180.

[3] Dumais, S. T. 1995. Latent Semantic Indexing (LSI): TREC-3 Report. In *Overview of the 3rd Text Retrieval Conference Gaithersburg, NIST*, pages 219-230.

[4] Kintsch, W. 1998. *Comprehension. A Paradigm for Cognition.* Cambridge University Press, Cambridge.

[5] Landauer, T. K. and S. T. Dumais. 1997. A Solution to Plato's Problem. *Psychological Review*, 104(2): 211-240.

[6] Lin, D. (1998): Automatic Retrieval and Clustering of Similar Words. *Proceedings of COLING-ACL '98*, 768-774.

[7] Mehler, A. (1998): Toward Computational Aspects of Text Semiotics. *Proceedings of IEEE ISIC/CIRA/ISAS*. Piscataway: IEEE/Omnipress, 807-813.

[8] Mehler, A. (2001): Aspects of Text Mining. From Computational Semiotics to Systemic Functional Hypertexts *Australian Journal of Information Science* 8(2), 129-141.

[9] Rieger, B. 1991. On Distributed Representation and Word Semantics. Technical Report 91-012, UC Berkeley, ICSI.

[10] Salton, G., Allan, J. and C. Buckley. 1994. Automatic Structuring and Retrieval of Large Text Files. *Commun. ACM*, 37(2): 97-108.

[11] Schütze, H. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1): 97-123.

[12] Sharkey, A. J. C. and N. E. Sharkey. 1992. Weak Contextual Constraints in Text and Word Priming. *J. Mem. Lang.*, 31(4): 543-572.

Contact

Dr. Alexander Mehler
Department of Linguistic Data Processing
Universität Trier
Universitätsring 15
D-54286 Trier, Germany
Email: `mehler@uni-trier.de`

Alexander Mehler studied linguistic data processing and economic science at the University of Trier. From 1995 to 1998 he headed a department for software development in industry. In November 2000 he received his PhD. Since 1998 he is scientific assistant of the Chair for Linguistic Data Processing at the University of Trier. His research interests include text linguistics, text mining, and computational semiotics.