Cognitive Approach to Natural Language Processing

Edited by Bernadette Sharp
Florence Sèdes and Wiesław Lubaszewski

ISTE PRESS
ELSEVIER

Cognitive Approach to Natural Language Processing

This page intentionally left blank

Series Editor
Florence Sèdes

Cognitive Approach to Natural Language Processing

Edited by

Bernadette Sharp
Florence Sèdes
Wiesław Lubaszewski

ISTE PRESS
ELSEVIER

First published 2017 in Great Britain and the United States by ISTE Press Ltd and Elsevier Ltd

ISTE Press Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

Elsevier Ltd
The Boulevard, Langford Lane
Kidlington, Oxford, OX5 1GB
UK

www.elsevier.com

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
Library of Congress Cataloging in Publication Data
A catalog record for this book is available from the Library of Congress
ISBN 978-1-78548-253-3

Printed and bound in the UK and US

Contents

Preface

This book is a special issue dedicated to exploring the relationship between natural language processing and cognitive science, and the contribution of computer science to these two fields. Poibeau and Vasishth [POI 16] noted that research interest in cognitive issues may have been given less attention because researchers from the cognitive science field are overwhelmed by the technical complexity of natural language processing; similarly, natural language processing researchers have not recognized the contribution of cognitive science to their work. We believe that the international workshops of Natural Language and Cognitive Science (NLPCS), launched in 2004, have provided a strong platform to support the consistent determination and diversity of new research projects which acknowledge the importance of interdisciplinary approaches and bring together computer scientists, cognitive and linguistic researchers to advance research in natural language processing.

This book consists of 10 chapters contributed by the researchers at the recent NLPCS workshops. In Chapter 1, Philippe Blache explains that the process of understanding language is theoretically very complex; it must be carried out in real time. This process requires many different sources of information. He argues that the global interpretation of a linguistic input is based on the grouping of elementary units called chunks which constitute the backbone of the "interpret whenever possible" principle which is responsible for delaying the understanding process until enough information becomes available. The following two chapters address the problem of human association. In Chapter 2, Korzycki, Gatkowska and Lubaszewski discuss an experiment based on 900 students who participated in a free word

association test. They have compared the human association list with the association list retrieved from text using three algorithms: the Church and Hanks algorithm, the Latent Semantic Analysis and Latent Dirichlet Allocation. In Chapter 3, Lubaszewski, Gatkowska and Godny describe a procedure developed to investigate word associations in an experimentally built human association network. They argue that each association is based on the semantic relation between two meanings, which has its own direction and is independent from the direction of other associations. This procedure uses graph structures to produce a semantically consistent subgraph. In Chapter 4, Rapp investigates whether human language generation is governed by associations, and whether the next content word of an utterance can be considered as an association with the representations of the content words, already activated in the speaker's memory. He introduces the concept of the Reverse Association Task and discusses whether the stimulus can be predicted from the responses. He has collected human data based on the reverse association task, and compared them to the machine-generated results. In Chapter 5, Vincent-Lamarre and his colleagues have investigated how many words, and which ones, are required to define all the rest of the words in a dictionary. To this end, they have applied graph-theoretic analysis to the Wordsmyth suite of dictionaries. The results of their study have implications for the understanding of symbol grounding and the learning and mental representation of word meaning. They conclude that language users must have the vocabulary to understand the words in definitions to be able to learn and understand the meaning of words from verbal definitions. Chapter 6 focuses on word sense disambiguation. Tripodi and Pelillo have explored the evolutionary game theory approach to study word sense disambiguation. Each word to be disambiguated is represented as a player and each sense as a strategy. The algorithm has been tested on four datasets with different numbers of labeled words. It exploits relational and contextual information to infer the meaning of a target word. The experimental results demonstrate that this approach has outperformed conventional methods and requires a small amount of labeled points to outperform supervised systems. In Chapter 7, Zock and Tesfaye have focused on the challenging task of text production expressed in terms of four tasks: ideation, text structuring, expression and revision. They have focused on text structuring which involves the grouping (chunking), ordering and linking of messages. Their aim is to study which parts of text production can be automated, and whether the computer can build one or several topic trees based on a set of inputs provided by the user. Authorship attribution is the focus of study in

Chapter 8. Boukhaled and Ganascia have analyzed the effectiveness of using sequential rules of function words and Part-of-Speech (POS) tags as a style marker that does not rely on the bag-of-words assumption or on their raw frequencies. Their study has shown that the frequencies of function words and POS n-grams outperform the sequential rules. Fundamental frequency detection (F0), which plays an important role in human speech perception, is addressed in Chapter 9. Glavitsch has investigated whether F0 estimation, using the principles of human cognition, can perform equally well or better than state-of-the-art F0 detection algorithms. The proposed algorithm, which operates in the time domain, has achieved very low error rates and outperformed the state-of-the-art correlation-based method RAPT in this respect, using limited resources in terms of memory and computing power. In neurocognitive psychology, manually collected cloze completion probabilities (CCPs) are used to quantify the predictability of a word from sentence context in models of eye movement control. As these CCPs are based on samples of up to 100 participants, it is difficult to generalize a model across all novel stimuli. In Chapter 10, Hofmann, Biemann and Remus have proposed applying language models which can be benchmarked by item-level performance on datasets openly available in online databases. Previous neurocognitive approaches to word predictability from sentence context in electroencephalographic (EEG) and eye movement (EM) data relied on cloze completion probability (CCP) data. Their study has demonstrated that the syntactic and short-range semantic processes of n-gram language models and recurrent neural networks (RNN) can perform more or less equally well when directly accounting CCP, EEG and EM data. This may help generalize neurocognitive models to all possible novel word combinations.

Bibliography

[POI 16] POIBEAU T., VASISHTH S., "Introduction: Cognitive Issues in Natural Language Processing", *Traitement Automatique des Langues et Sciences Cognitives*, vol. 55, no. 3, pp. 7–19, 2016.

Bernadette SHARP
Florence SÈDES
Wiesław LUBASZEWSKI
March 2017

This page intentionally left blank

1

Delayed Interpretation, Shallow Processing and Constructions: the Basis of the "*Interpret Whenever Possible*" Principle

We propose in this chapter to investigate the "*interpret whenever possible*" principle that consists of delaying the processing mechanisms until enough information becomes available. This principle relies on the identification of elementary units called chunks, which are identified by means of basic features. These chunks are segments of the input to be processed. In some cases, depending on the accessibility of the information they bear, chunks can be linguistically structured elements. In other cases, they are simple segments. Chunks are stored in a buffer of the working memory and progressively grouped (on the basis of a cohesion measure) when possible, progressively identifying the different constructions of the input. The global interpretation of a linguistic input is then not based anymore on a word-by-word mechanism, but on the grouping of these constructions that constitute the backbone of the "*interpret whenever possible*" principle.

1.1. Introduction

From different perspectives, natural language processing, linguistics and psycholinguistics shed light on the way humans process language. However, this knowledge remains scattered: classical studies usually focus on language processing subtasks (e.g. lexical access) or modules (e.g. morphology,

---

Chapter written by Philippe BLACHE.

syntax), without being aggregated into a unified framework. It then remains very difficult to find a general model unifying the different sources of information into a unique architecture.

One of the problems lies in the fact that we still know only little about how the different dimensions of language (prosody, syntax, pragmatics, semantics, etc.) interact. Some linguistic theories exist, in particular within the context of *Construction Grammars* [FIL 88, GOL 03, BLA 16], that propose approaches making it possible to gather these dimensions and implement their relations. These frameworks rely on the notion of *construction*, which is a set of words linked by specific properties at any level (lexical, syntactic, prosodic, etc.) and with which a specific meaning, which is often non-transparent or accessible compositionally (e.g. idioms or multi-word expressions), can be associated. Interestingly, these theories also provide a framework for integrating multimodal information (verbal and non-verbal). Interpreting a construction (i.e. accessing to its associated meaning) results from the interaction of all the different dimensions. In this organization, processing a linguistic production is not a linear process, but uses mechanisms for a global recognition of the constructions. Contrarily to incremental architectures (see, e.g., [FER 02, RAY 09]), the syntactic, semantic and pragmatic processing is not done word-by-word, but more globally, on the basis of such constructions.

This conception of language processing requires a *synchronization* procedure for the alignment of all the different sources of information in order to identify a construction and access its meaning. In natural situations (e.g. conversations), the different input flows can be verbal (prosody, syntactic, pragmatics, etc.) and non-verbal (gestures, attitudes, emotions, context, etc.); they are not strictly temporally synchronized. It is then necessary to explain how information can be temporarily stored and its evaluation delayed until enough information becomes available. In this perspective, the input linguistic flow (being read or heard) is segmented into elements that can be of any form, partially or entirely recognized: segments of the audio flow, set of characters, and also, when possible, higher-level segments made of words or even clusters of words. In this chapter, we address these problems through several questions:

1) What is the nature of the delaying mechanism?

2) What is the nature of the basic units and how can they be identified?

3) How is the delaying mechanism implemented?

1.2. Delayed processing

Different types of delaying effects can occur during language processing. For example, at the brain level, it has been shown that language processing may be impacted by the presentation rate of the input. This phenomena has been investigated in [VAG 12], claiming that when the presentation rate increases and becomes faster than the processing speed, intelligibility can collapse. This is due to the fact that language network seems to work in a constant of time: cortical processing speed is shown by the authors to be tightly constrained and cannot be easily accelerated. As a result, when the presentation rate increases, the processing speed remaining constant, a blocking situation can suddenly occur. Concretely, this means that when the presentation rate is accelerated, and because the processing speed remains constant, part of the input stream has to be buffered. Experiments show that the rate can be accelerated to 40% before reaching a collapse of intelligibility. This situation occurs when the buffer becomes saturated and is revealed at the cortical level by the fact that the activation of the higher-order language areas (that are said to reflect intelligibility [FRI 10]) drops suddenly, showing that the input signal becomes unintelligible.

This model suggests that words can be processed immediately when presented at a slow rate, in which case the processing speed is that of the sensory system. However, when the rate increases and words are presented more rapidly, the processing speed limit is reached and words cannot be processed in real time anymore. In such a situation, words have to be stored in a buffer, from which they are retrieved in a *first-in-first-out* manner, when cognitive resources become available again. When the presentation rate is higher than the processing speed, the number of words to be stored increases. A lock occurs when the maximal capacity of the buffer is reached, entailing a collapse of intelligibility.

Besides this buffering mechanism, other cues indicate that the input is probably not processed linearly, word-by-word, but rather only from time to time. This conception means that even in normal cases (i.e. without any intelligibility issue), the interpretation is only done periodically, the basic units being stored before being processed. Several studies have investigated such a phenomenon. At the cortical level, the analysis of stimulus intensity fluctuation reveals the presence of specific activity (spectral peaks) after phrases and sentences [DIN 16]. The same type of effect can also be found in

eye-movement during reading: longer fixations are observed when reading words that end a phrase or a sentence. This *wrap-up effect* [WAR 09], as well as the presence of different timescales at the cortical level described above, constitute cues in favor of a delaying mechanism in which basic elements are stored temporarily, and an integration operation is triggered when enough material becomes available for the interpretation.

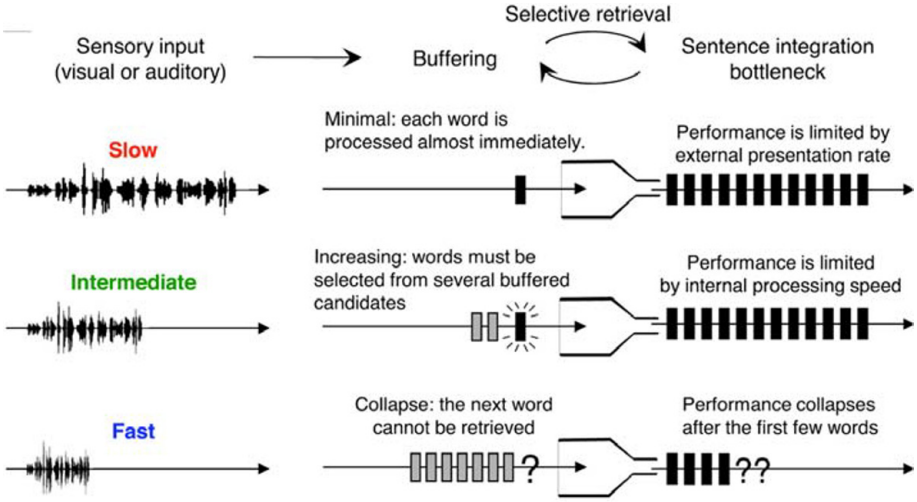


Figure 1.1. *Illustration of the bottleneck situation, when the presentation rate exceeds the processing speed (reproduced from [VAG 12])*

At the semantic level, other evidence also shows that language processing, or at least language interpretation, is not strictly incremental. Interesting experiments have been performed, which reveal that language comprehension can remain very superficial: [ROM 13] has shown that, in an idiomatic context, the access to the meaning of words can be completely switched off, replaced by a global access at the level of idiom. This effect has been shown at the cortical level: when introducing a semantic violation within an idiom, there is no difference between hard and soft semantic violations (which is not the case in a comparable non-idiomatic context); in some cases, processing a word does not mean integrating it into a structure. On the contrary, in this situation there is a simple shallow process of scanning the word, without

doing any interpretation. The same type of observation has been made in reading studies: depending on the task (e.g. when very simple comprehension questions are expected), the reader may apply a superficial treatment [SWE 08]. This effect is revealed by the fact that ambiguous sentences are read faster, meaning that no resolution is done and the semantic representation remains underspecified. Such variation in the level of processing depends then on the context: when the pragmatic and semantic context carries enough information, it renders the complete processing mechanism useless, the interpretation being predictable. At the attentional level, this observation is confirmed in [AST 09], showing that the allocation of attentional resources to certain time windows depends on its predictability: minimal attention is allocated when information is predictable or, on the contrary, maximal attention is involved in case of mismatch with expectations. The same type of variation is observed when the listener adapts its perceptual strategy to the speakers, applying *perceptual accommodation* [MAG 07].

These observations are in line with the *good-enough theory* [FER 07] for which the interpretation of complex material is often considered to be shallow and incomplete. This model suggests that interpretation is only done from time to time, on the basis of a small number of adjacent words, and delaying the global interpretation until enough material becomes available. This framework and the evidence on which it relies also reinforce the idea that language processing is generally not linear and word-by-word. On the contrary, it can be very shallow and delayed when necessary.

1.3. Working memory

The delaying mechanism relies implicitly on a storage device which is implemented in the *short-term memory*, which is the basis of the cognitive system organization, by making it possible to temporarily store pieces of information of any nature. In general, it is considered that this memory is mainly devoted to storage. However, a specific short-term memory, called *working memory*, also allows for the manipulation of the information and a certain level of processing. It works as a buffer in which part of the information, which can be partially structured, is stored. Some models [BAD 86, BAD 00] propose an architecture in which the working memory plays the role of a supervisor, on top of different sensory-motor loops as well as an episodic buffer.

One important feature of the working memory (and short-term memory in general) is its limited capacity. In a famous paper, [MIL 56] evaluated this limit to a "magic" number of seven units. However, it has been observed that units to be stored in this memory are not necessarily atomic; they can also constitute groups that are considered as single units. For example, stored elements can be numbers, letters, words, or even sequences, showing that groups can be encoded as single units. In this case, the working memory does not directly store the set of elements, but more probably the set of pointers towards the location of the elements in another (lower) part of the short-term memory. These types of higher-level elements are called *chunks*, which basically consist, in the case of language, of a set of words.

Working memory occupies a central position in cognitive architectures such as ACT-R (*Adaptive Character of Thought-Rational*, see [AND 04]). In this model, short-term information (chunks) is stored in a set of buffers. The architecture, in the manner of that proposed by [BAD 86], is organized around a set of modules (manual control, visual perception, problem state, control state and declarative memory) coordinated by a supervising system (the production system). Each module is associated with a buffer that contains one chunk, defined as a unit containing a small amount of information. Moreover, in this organization, each buffer can contain only one unit of knowledge.

ACT-R has been applied to language processing, in which short-term buffers play the role of an interface between procedural and declarative memories (the different types of linguistic knowledge) [LEW 05, REI 11]. Buffers contain chunks (information units) that are represented as lists of attribute-value pairs. Chunks are stored in the memory, they form a unit and they can be directly accessible, as a whole. Their accessibility depends on a level of *activation*, making it possible to control their retrieval in the declarative memory. A chunk's activation consists of several parameters: latency since its last retrieval, weights of the elements in relation to the chunk as well as the strength of these relations. It can be integrated into the following formula, quantifying the activation A of a chunk i:

$$A_i = B_i + \sum_j W_j S_{ji} \qquad [1.1]$$

In this formula, B represents the basic activation of the chunk (its frequency and the recency of its retrieval), W indicates the weights of the

terms in relation to i and S the strength of the relations linking other terms to the chunks. It is then possible to associate a chunk with its level of activation. The interesting point is that chunk activation is partially dependent on the context: the strength of the relations with other elements has a consequence on the level of activation, controlling its probability as well as the speed of its retrieval.

This architecture implicitly contains the idea of delayed evaluation: the basic units are first identified and stored into different buffers, containing pieces of information that can be atomic or structured. Moreover, this proposal also gives indications on the type of the retrieval. The different buffers in which chunks are stored are not implemented as a stack, following a *first-in-first-out* retrieval mechanism. On the contrary, chunks can be retrieved in any order, with a preference given first to that with the higher activation value.

The ACT-R model and the activation notion give a more precise account of comprehension difficulties. In the previous section, we have seen that they can be the consequence of a buffer saturation (in computational terms, a *stack overflow*). Such difficulties are controlled thanks to the *decay of accessibility* of stored information [LEW 05]. This explanation is complementary with observations presented in the previous section: the activation level has a correlation with the processing speed. Chunks with a high activation will be retrieved rapidly, decreasing the number of buffered elements. When many chunks have a low activation, the processing speed decreases, resulting in a congestion of the buffers.

One important question in this architecture is the role of working memory in procedural operations, and more precisely the construction of the different elements to be stored. In some approaches, working memory plays a decisive role in terms of integration: basic elements (lexical units) are assembled into structured ones, as a function of their activation. In this organization, working memory becomes the site where linguistic analysis is done. This is what has been proposed, for example, in the *"Capacity theory of comprehension"* [JUS 92] for which working memory plays a double role of storage and processing. In this theory, elements of any level can be stored and accessed: words, phrases, thematic structures, pragmatic information, etc. It is, however, difficult to explain how such model can implement at the same time

a delaying aspect (called "*wait-and-see*" by the authors) and an incremental comprehension system interpreting step-by-step. In their study on memory capacity, [VAG 12] propose a simpler view with a unique input buffer whose role is limited to storing words. In our approach, we adopt an intermediate position in which the buffer is limited to storage, but elements of different types can be stored, including partially structured ones such as chunks.

1.4. How to recognize chunks: the segmentation operations

The hypothesis of a delayed evaluation in language processing not only relies on a specific organization of the memory, but also requires a mechanism for the identification of the elements to be stored in the buffer. Two important questions are to be answered here: what is the nature of these elements, and how can they be identified. Our hypothesis relies on the idea that no deep and precise linguistic analysis is done at a first stage. If so, the question is to explain and describe the mechanisms, necessarily at a low level, for the identification of the stored elements.

These questions are more generally related to the general problem of segmentation. Given an input flow (e.g. connected speech), what types of element can be isolated and how? Some mechanisms, specific to the audio signal, are at work in speech segmentation. Many works addressing this question ([MAT 05], [GOY 10], [NEW 11], [END 10]) exhibit different cues, at different levels, that are used in particular (but not only) for word segmentation tasks, among which:

– *Prosodic level*: stress, duration and pitch information can be associated in some languages with specific positions in the word (e.g. initial or final), helping in detecting the word boundaries.

– *Allophonic level*: phonemes are variable and their realization can depend on their position within words.

– *Phonotactic level*: constraints on the ordering of the phonemes, which gives information about the likelihood that a given phoneme is adjacent to another one within and between words.

– *Statistical/distributional properties*: transitional probabilities between consecutive syllables.

Word segmentation results from the satisfaction of multiple constraints encoding different types of information, such as phonetic, phonological, lexical, prosodic, syntactic, semantic, etc. (see [MCQ 10]). However, most of these segmentation cues are at a low level and do not involve an actual lexical access. In this perspective, what is interesting is that some segmentation mechanisms are not dependent on the notion of word and then can also be used in other tasks than word segmentation. This is very important because the notion of word is not always relevant (because involving rather high-level features, including semantic ones). In many cases, other types of segmentations are used, without involving the notion of words, but staying at the identification of larger segments (e.g. prosodic units), without entering into a deep linguistic analysis.

At a higher level, [DEH 15] has proposed isolating five mechanisms making it possible to identify sequence knowledge:

– *Transition and timing knowledge*: when presenting a sequence of items (of any nature), at a certain pace, the transition between two items is anticipated thanks to the approximate timing of the next item.

– *Chunking*: contiguous items can be grouped into the same unit, thanks to the identification of certain regularities. A chunk is simply defined here in terms of a set of contiguous items that frequently co-occur and then can be encoded as a single unit.

– *Ordinal knowledge*: a recurrent linear order, independently of any timing, constitutes information for the identification of an element and its position.

– *Algebraic patterns*: when several items have an internal regular pattern, their identification can be done thanks to this information.

– *Nested tree structures generated by symbolic rules*: identification of a complex structure, gathering several items into a unique element (typically a phrase).

What is important in these sequence identification systems (at least the first four of them) is the fact that they apply to any type of information and rely on low-level mechanisms, based on the detection of regularities and when possible their frequency. When applied to language, these systems explain how syllables, patterns or groups can be identified directly. For

example, algebraic patterns are specific to a certain construction such as in the following example, taken from a spoken language corpus: "*Monday, washing, Tuesday, ironing, Wednesday, rest*". In this case, without any syntactic or high-level processing, and thanks to the regularity of the pattern /*date - action*/, it is possible to segment the three subsequences and group them into a unique general one. In this case, a very basic mechanism, *pattern identification*, offers the possibility to identify a construction (and access directly to its meaning).

When putting together the different mechanisms described in this section, we obtain a strong set of parameters that offer the possibility of segmenting the input into units. In some cases, when cues are converging enough, the segments can be words. In other cases, they are larger units. For example, long breaks (higher than 200ms) are a universal segmentation constraint in prosody: two such breaks identify the boundaries of a segment (that can correspond to a prosodic unit).

As a result, we can conclude that several basic mechanisms, which do not involve deep analysis, make it possible to segment the linguistic input, be it read or heard. Our hypothesis is that these segments are the basic units stored initially in the buffers. When possible, the stored units are words, but not necessarily. In the general case, they are sequences of characters or phonemes that can be retrieved later. This is what occurs when hearing a speaker without understanding: the audio segment is stored and accessed later when other sources of information (e.g. the context) become available and make it possible to refine the segmentation into words.

1.5. The delaying architecture

Following the different elements presented so far, we propose integrating the notion of delayed evaluation and chunking into the language processing organization. This architecture relies on the idea that the interpretation of a sentence (leading to its comprehension) is only done *whenever possible*, instead of word-by-word. The mechanism consists of accumulating enough information before any in-depth processing. Doing this means: first, the capacity to identify atomic units without making use of any deep parsing, and; second, to store these elements and retrieve them when necessary.

We do not address here the question of building an interpretation, but focus only on this preliminary phase of accumulating pieces of information. This organization relies on a two-stage process distinguishing between a first level of packaging and a second corresponding to a deeper analysis. Such a distinction recalls the well-known "*Sausage Machine*" [FRA 78] that distinguishes a first phase called the *Preliminary Phrase Packager (PPP)*, consisting of identifying the possible groups (or chunks) in a limited window made of 6 or 7 words. In this proposal, the groups correspond to phrases that can be incomplete. The second level is called the *Sentence Structure Supervisor (SSS)* and it groups the units produced in the *PPP* into larger structures. In this classical architecture, each level involves a certain type of parsing, relying on grammatical knowledge. Moreover, the interpretation is supposed to be done starting from the identification of the syntactic structure, in a classical compositional perspective.

Our proposal also relies on a two-stage organization:

1) segmenting and storing;

2) aggregating complex chunks.

However, this model does not have any *a priori* on the type of units to be built: they are not necessarily phrases, they can be simply made of unstructured segments of the input. Moreover, the second stage is not obligatory: the recognition of a construction, and the interpretation of the corresponding subpart of the input, can be done at the first level.

We detail in the following these two stages, on the basis of the more general "*interpretation whenever possible*" organization.

1.5.1. *Segment-and-store*

The first stage when processing a linguistic input (text or speech) is the segmentation into atomic chunks. Atomic means here that no structure is built, chunks being only segments of the input, identified thanks to low-level parameters. In other words, no precise analysis of the input is performed, the mechanism consisting of gathering all possible information available immediately. As a result, because the level of precision of the information

can be very different, chunks can be of many different types and levels. Some of the segmentation mechanisms are indeed very general or even universal. For example, the definition of "*inter-pausal units*" relies on the identification of long breaks in the audio signal. The resulting chunk is a long sequence of phonemes without internal organization or sub-segmentation. In some (rare) cases, no other features than long breaks are available and the chunk remains large and stored as such. However, in most of the situations, more information is available, making it possible to identify finer chunks, and when possible words. Several such segmenting features exist, in particular:

– *Prosodic contours, stress*: pitch, breaks, duration and stress may indicate word boundaries.

– *Phonotactic constraints*: language-dependent constraints on the sequence of phonemes. The violation of such constraints may indicate boundaries.

– *Lexical frequency units*: in some cases, an entire unit can be highly predictable (typically very frequent words, named entities, etc.), making it possible to directly segment the input.

These features are subject to high variation and do not lead to a segmentation in all cases. When ambiguity is high, no finer segmentation is done at this stage. On the contrary, these low-level features can often lead to the possibility of segmenting into words. What is important is that these features correspond to information that can be directly assessed, independently of any other property or knowledge.

At this first stage, atomic chunks are stored into the buffers. We present in the following section the next step of this pre-processing phase, consisting of aggregating chunks.

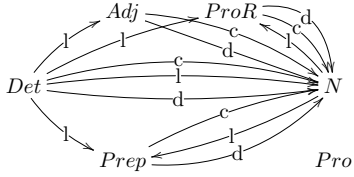
1.5.2. *Aggregating by cohesion*

Constructions can be described as a set of interacting properties. This definition offers the possibility to conceive a measure based on the number of these properties and their weights, as proposed in [BLA 16]. At the syntactic level, the set of properties describing a construction corresponds to a graph in which nodes are words and edges represent the relations. The graph density

then constitutes a first type of measure: a high density of the graph corresponds to a high number of properties, representing a certain type of cohesion between the words. Moreover, the quality of these relations can also be evaluated, some properties being more important than others (which are represented by their weighting). A high density of hard properties (i.e. with heavy weights) constitute a second type of information. Finally, some sentences can be non-canonical, bearing certain properties that are violated (e.g. in case of agreement or linear precedence violation). Taking into consideration the number of violated properties in comparison with the satisfied ones is the last type of indication we propose to use in the evaluation of the cohesion.

Our hypothesis is that a correlation exists between the cohesion measure, defined on the basis of these three types of information, and the identification of a construction. In other words, a construction corresponds to a set of words linked with a high number of properties, of heavy weights, with no or few violations.

The first parameter of the cohesion measure relies on the number of properties that are assessed for a given construction, in comparison with the possible properties in the grammar. The following graph illustrates the set of properties *in the grammar* describing the nominal construction[1]:

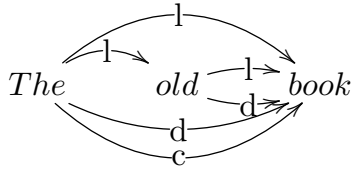


[1.2]

The number of possible relations in which a category is involved can be estimated by the number of incident relations of the corresponding vertex in the graph (called in graph theory the *vertex* degree). We then propose to define the degree of a category by this measure. In the previous graph, we have the following degrees: $deg_{[gram]}(N) = 9; deg_{[gram]}(ProR) = 2;$ $deg_{[gram]}(Adj) = 1.$

1 The letters d, l, c stand respectively for *dependency, linearity* and *co-occurrence* properties.

During a parse (i.e. knowing the list of categories), the same type of evaluation can be applied to the constraint graph describing a construction, as in the following example:



Each word is involved in a set of relations. The degree of a word is, similarly to the grammar, the set of incident edges of a word. In this example, we have: $deg_{[sent]}(N) = 5; deg_{[sent]}(Adj) = 1; deg_{[sent]}(Det) = 0$.

The first parameter of our estimation of the cohesion relies on a comparison of these two values: for a given word, we know from the grammar the number of properties in which it could theoretically be involved. We also know from the parsing of a given sentence how many of these properties are effectively assessed. We can then define a value, the *completeness ratio*, indicating the density of the category: the higher the number of relations in the grammar being verified, the higher the completeness value:

$$Comp(cat) = \frac{deg_{[sent]}(cat)}{deg_{[gram]}(cat)} \quad [1.3]$$

Besides this completeness ratio, it is also interesting to examine the density of the constraint graph itself. In graph theory, this value is calculated as a ratio between the number of edges and the number of vertices. It is more precisely defined as follows (S is the constraint graph of a sentence, E the set of edges and V the set of vertices):

$$Dens(S) = \frac{|E|}{5 * |V|(|V| - 1)} \quad [1.4]$$

In this formula, the numerator is the number of existing edges and the denominator is the total number of possible edges (each edge connecting two different vertices, multiplied by 5, the number of different types of properties). This value makes it possible to distinguish between *dense* and

sparse graphs. In our hypothesis, a dense graph is correlated with a construction.

The last parameter taken into account is more qualitative and takes into account the weights of the properties. More precisely, we have seen that all properties can be either satisfied or violated. We define then a normalized satisfaction ratio as follows (where W^+ is the sum of the weights of the satisfied properties and W^- that of the violated ones):

$$Sat(S) = \frac{W^+ - W^-}{W^+ + W^-} \quad [1.5]$$

Finally, the cohesion value can be calculated as a function of the three previous parameters as follows (C being a construction and G_C its corresponding constraint graph):

$$Cohesion(C) = \sum_{i=1}^{|S|} Comp(w_i) * Dens(G_C) * Sat(G_C) \quad [1.6]$$

Note that the *density* and *satisfaction* parameters can be evaluated directly, without depending on the context and without needing to know the type of the construction. On the contrary, evaluating the *completeness* parameter requires knowing the construction in order to extract from the grammar all the possible properties that describe it. In a certain sense, the two first parameters are *basic*, in the same sense as described for properties, and can be assessed automatically.

The *cohesion* measure offers a new estimation of the notion of *activation*. Moreover, it also provides a way to directly identify constructions on the basis of simple properties. Finally, it constitutes an explicit basis for the implementation of the general parsing principle stipulating that constructions or chunks are set of words with a high density of relations of heavy weights. This definition corresponds to the *Maximize On-line Processing* principle [HAW 03], which stipulates that *"the human parser prefers to maximize the set of properties that are assignable to each item X as X is parsed. [...] The maximization difference between competing orders and structures will be a function of the number of properties that are misassigned or unassigned to X in a structure S, compared with the number in an alternative"*.

This principle offers a general background to our conception of language processing. Instead of building a syntactic structure serving as support of the comprehension of a sentence, the mechanisms consist of a succession of chunks, maximizing the cohesion function estimated starting from the available information. When the density of information (or the cohesion) reaches a certain threshold, the elements can be grouped into a unique chunk, stored in the working memory. When the threshold is not reached, the state of the buffer is not modified and a new element of the input stream is scanned. This general parsing mechanism offers the possibility to integrate different sources of information when they become available by delaying the evaluation, waiting until a certain threshold of cohesion can be identified. This constitutes a framework for implementing the basic processing of the good-enough theory: *interpret whenever possible*.

1.6. Conclusion

Understanding language is theoretically a very complex process, involving many different sources of information. Moreover, it has to be done in real time. Fortunately, in many cases, the understanding process can be facilitated thanks to different parameters: predictability of course, and also the fact that entire segments of the input can be processed directly. This is the case of most of the *constructions*, in which the meaning can be accessed directly, the construction being processed as a whole. At a lower level, it is also possible to identify subparts of the input (e.g. patterns, prosodic units, etc.) from which global information can be retrieved directly. Different observations show that low-level features usually make it possible to identify such global segments. The language processing architecture we propose in this chapter relies on this: instead of recognizing words and then trying to integrate them step-by-step into a syntactic structure to be interpreted, segments are first identified. These segments can be of any type: sequences of phonemes, words, group of words, etc. Their common feature is that they do not need any deep level information or process to be recognized.

Once the segments (called *chunks*) are identified, they are stored in a buffer, without any specific interpretation. In other words, the interpretation mechanism is *delayed* until enough information becomes available. When a new chunk is buffered, an evaluation of its *cohesion* with the existing ones in the buffer is done. When the cohesion between different chunks (that

corresponds to the notion of activation in cognitive architectures) reaches a certain threshold, they are merged into a unique one, replacing them in the buffer, as a single unit. This mechanism makes it possible to progressively recognize constructions and directly access their meaning.

This organization, instead of a word-by-word incremental mechanism, implements the *"interpret whenever possible"* principle. It constitutes a framework for explaining all the different delaying and shallow processing mechanisms that have been observed.

1.7. Bibliography

[AND 04] ANDERSON J.R., BOTHELL D., BYRNE M.D. *et al.*, "An integrated theory of the mind", *Psychological Review*, vol. 111, no. 4, pp. 1036–1060, 2004.

[AST 09] ASTHEIMER L.B., SANDERS L.D., "Listeners modulate temporally selective attention during natural speech processing", *Biological Psychology*, vol. 80, no. 1, pp. 23–34, 2009.

[BAD 86] BADDELEY A., *Working Memory*, Clarendon Press, Oxford, 1986.

[BAD 00] BADDELEY A., "The episodic buffer: a new component of working memory?", *Trends in Cognitive Sciences*, vol. 4, no. 11, pp. 417–423, 2000.

[BLA 16] BLACHE P., "Representing syntax by means of properties: a formal framework for descriptive approaches", *Journal of Language Modelling*, vol. 4, no. 2, 2016.

[DEH 15] DEHAENE S., MEYNIEL F., WACONGNE C. *et al.*, "The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees", *Neuron*, vol. 88, no. 1, 2015.

[DIN 16] DING N., MELLONI L., ZHANG H. *et al.*, "Cortical tracking of hierarchical linguistic structures in connected speech", *Nature Neuroscience*, vol. 19, no. 1, pp. 158–164, 2016.

[END 10] ENDRESS A.D., HAUSER M.D., "Word segmentation with universal prosodic cues", *Cognitive Psychology*, vol. 61, no. 2, pp. 177–199, 2010.

[FER 02] FERREIRA F., SWETS B., "How incremental is language production? Evidence from the production of utterances requiring the computation of arithmetic sums", *Journal of Memory and Language*, vol. 46, no. 1, pp. 57–84, 2002.

[FER 07] FERREIRA F., PATSON N.D., "The 'Good Enough' approach to language comprehension", *Language and Linguistics Compass*, vol. 1, no. 1, 2007.

[FIL 88] FILLMORE C.J., "The mechanisms of 'Construction Grammar"', *Proceedings of the Fourteenth Annual Meeting of the Berkeley Linguistics Society*, pp. 35–55, 1988.

[FRA 78] FRAZIER L., FODOR J.D., "The sausage machine: a new two-stage parsing model", *Cognition*, vol. 6, no. 4, pp. 291–325, 1978.

[FRI 10] FRIEDERICI A., KOTZ S., SCOTT S. *et al.*, "Disentangling syntax and intelligibility in auditory language comprehension", *Human Brain Mapping*, vol. 31, no. 3, pp. 448–457, 2010.

[GOL 03] GOLDBERG A.E., "Constructions: a new theoretical approach to language", *Trends in Cognitive Sciences*, vol. 7, no. 5, pp. 219–224, 2003.

[GOY 10] GOYET L., DE SCHONEN S., NAZZI T., "Words and syllables in fluent speech segmentation by French-learning infants: an ERP study", *Brain Research*, vol. 1332, no. C, pp. 75–89, 2010.

[HAW 03] HAWKINS J., "Efficiency and complexity in grammars: three general principles", in MOORE J., POLINSKY M. (eds), *The Nature of Explanation in Linguistic Theory*, CSLI Publications, 2003.

[JUS 92] JUST M.A., CARPENTER P.A., "A capacity theory of comprehension: individual differences in working memory", *Psychological Review*, vol. 99, no. 1, pp. 122–149, 1992.

[LEW 05] LEWIS R.L., VASISHTH S., "An activation-based model of sentence processing as skilled memory retrieval", *Cognitive Science*, vol. 29, pp. 375–419, 2005.

[MAG 07] MAGNUSON J., NUSBAUM H., "Acoustic differences, listener expectations, and the perceptual accommodation of talker variability", *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, no. 2, pp. 391–409, 2007.

[MAT 05] MATTYS S.L., WHITE L., MELHORN J.F., "Integration of multiple speech segmentation cues: a hierarchical framework", *Journal of Experimental Psychology*, vol. 134, no. 4, pp. 477–500, 2005.

[MCQ 10] MCQUEEN J.M., "Speech perception", in LAMBERTS K., GOLDSTONE R. (eds), *The Handbook of Cognition*, Sage, London, 2010.

[MIL 56] MILLER G., "The magical number seven, plus or minus two: some limits on our capacity for processing information", *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[NEW 11] NEWMAN R.S., SAWUSCH J.R., WUNNENBERG T., "Cues and cue interactions in segmenting words in fluent speech", *Journal of Memory and Language*, vol. 64, no. 4, 2011.

[RAY 09] RAYNER K., CLIFTON C., "Language processing in reading and speech perception is fast and incremental: implications for event related potential research", *Biological Psychology*, vol. 80, no. 1, pp. 4–9, 2009.

[REI 11] REITTER D., KELLER F., MOORE J.D., "A computational cognitive model of syntactic priming", *Cognitive Science*, vol. 35, no. 4, pp. 587–637, 2011.

[ROM 13] ROMMERS J., DIJKSTRA T., BASTIAANSEN M., "Context-dependent semantic processing in the human brain: evidence from idiom comprehension", *Journal of Cognitive Neuroscience*, vol. 25, no. 5, pp. 762–776, 2013.

[SWE 08] SWETS B., DESMET T., CLIFTON C. *et al.*, "Underspecification of syntactic ambiguities: evidence from self-paced reading", *Memory and Cognition*, vol. 36, no. 1, pp. 201–216, 2008.

[VAG 12] VAGHARCHAKIAN L.G.D.-L., PALLIER C., DEHAENE S., "A temporal bottleneck in the language comprehension network", *Journal of Neuroscience*, vol. 32, no. 26, pp. 9089–9102, 2012.

[WAR 09] WARREN T., WHITE S.J., REICHLE E.D., "Investigating the causes of wrap-up effects: evidence from eye movements and E–Z reader", *Cognition*, vol. 111, no. 1, pp. 132–137, 2009.

This page intentionally left blank

2

Can the Human Association Norm Evaluate Machine-Made Association Lists?

This chapter presents a comparison of a word association norm created by a psycholinguistic experiment to association lists generated by algorithms operating on text corpora. We compare lists generated by the Church–Hanks algorithm and lists generated by the LSA algorithm. An argument is presented on how those automatically generated lists reflect semantic dependencies present in human association norm, and that future comparisons should take into account a deeper analysis of the human association mechanisms observed in the association list.

2.1. Introduction

For more than three decades, there has been a commonly shared belief that word occurrences retrieved from a large text collection may define the lexical meaning of a word. Although there are some suggestions that co-occurrences retrieved from texts [RAP 02, WET 05] reflect the text's contiguities, there also exist suggestions that algorithms, such as the LSA, are unable to distinguish between co-occurrences which are corpus-independent semantic dependencies (elements of a semantic prototype) and co-occurrences which are corpus-dependent factual dependencies [WAN 05, WAN 08]. We shall adopt the second view to show that existing statistical algorithms use mechanisms which improperly filter word co-occurrences retrieved from texts. To prove this supposition, we shall

Chapter written by Michał KORZYCKI, Izabela GATKOWSKA and Wiesław LUBASZEWSKI.

compare the human association list to the association list retrieved from a text by three different algorithms, i.e. the Church–Hanks [CHU 90] algorithm, the Latent Semantic Analysis (LSA) algorithm [DEE 90] and the Latent Dirichlet Allocation (LDA) algorithm [BLE 03].

LSA is a word/document matrix rank reduction algorithm, which extracts word co-occurrences from within a text. As a result, each word in the corpus is related to all co-occurring words and all texts in which it occurs. This makes a base for an associative text comparison. The applicability of the LSA algorithm is the subject of various types of research, which range from text content comparison [DEE 90] to the analysis of human association norm [ORT 12]. However, there is still little interest in studying the linguistic significance of machine-made associations.

It seems obvious that a comparison of the human association norm and machine-created association list should be the base of this study. And we can find some preliminary studies based on such a comparison: [WAN 05, WET 05, WAN 08], the results of which show that the problem needs further investigation. It is worth noting that all the types of research referred to used a stimulus–response association strength to make a comparison. The point is that, if we compare association strength computed for a particular stimulus–response pair in association norms for different languages, we can find that association strength differs, e.g. butter is the strongest (0.54) response to stimulus bread in the Edinburgh Associative Thesaurus (EAT), but in the Polish association norm described below the association chleb "bread"–masło "butter" is not the strongest (0.075). In addition, we can observe that association strength may not distinguish a semantic and non-semantic association, e.g. roof 0.04, Jack 0.02 and wall 0.01, which are responses to the stimulus "house" in EAT. Therefore, we decided to test machine-made association lists against human association norms excluding association strength. As a comparison, we use the norm made by Polish speakers during a free word association experiment [GAT 14], hereinafter referred to as the author's experiment. Because both LSA and LDA use the whole text to generate word associations, we also tested human associations against the association list generated by the Church–Hanks algorithm [CHU 90], which operates on a sentence-like text window. We also used three different text corpora.

2.2. Human semantic associations

2.2.1. *Word association test*

Rather early on, it was noted that words in the human mind are linked. American clinical psychologists G. Kent and A.J. Rosanoff [KEN 10] perceived the diagnostic usefulness of an analysis of the links between words. In 1910, the duo created and conducted a test of the free association of words. They conducted research on 1,000 people of varied educational backgrounds and professions, asking their research subjects to give the first word that came into their minds as the result of a stimulus word. The study included 100 stimulus words (principally nouns and adjectives). The Kent–Rosanoff list of words was translated into several languages, in which this experiment was repeated, thereby enabling comparative research to be carried out. Word association research was continued in [PAL 64], [POS 70], [KIS 73], [MOS 96], [NEL 98], and the repeatability of results allowed the number of research subjects to be reduced, while at the same time increasing the number of word stimuli to be employed, for example 500 kids and 1,000 mature research subjects and 200 words [PAL 64] or 100 research subjects and 8,400 words [KIS 73]. Research on the free association of words has also been conducted in Poland [KUR 67], the results of which are the basis for the experiment described below.

Computational linguistics also became involved in research on the free association of words, though at times these experiments did not employ the rigors used by psychologists when conducting experiments, for example, those that permitted the possibility of providing several responses to an individual stimulus word [SCH 12] or those that used word pairs as a stimulus [RAP 08].

There exist some algorithms which generate an association list on the basis of text corpora. However, automatically generated associations were rather reluctantly compared with the results of psycholinguistic experiments. The situation is changing; Rapp's results [RAP 02] were really encouraging.

Finally, association norms are useful for different tasks, for example information extraction [BOR 09] or dictionary expansion [SIN 04, BUD 06].

2.2.2. *The author's experiment*

Some 900 students of the Jagiellonian University and AGH University of Technology participated in a free word association test as described in this chapter. A Polish version of the Kent–Rosanoff list of stimulus words, which was previously used by I. Kurcz, was employed [KUR 67]. After an initial analysis, it was determined that we would employ as a stimulus, each word from the Kent–Rosanoff list, which grammatically speaking is a noun, as well as the five most frequent word associations for each of those nouns obtained in Kurcz's experiment [KUR 67]. If given associations appeared for various words, for example, *white* for *doctor*, *cheese*, *sheep*, that word as a stimulus appeared only once in our experiment. The resulting stimulus list contained 60 words from the Kent–Rosanoff list, in its Polish version, as well as 260 words representing those associations (responses) which most frequently appeared in Kurcz's research. It, therefore, is not an exact repetition of the experiment conducted 45 years ago.

The conditions of the experiment conducted, as well as the method of analyzing the results, have been modified. The experiment was conducted in a computer lab, with the aid of a computer system which was created specifically for the requirements of this experiment. This system presents a list of stimuli and then stores the associations in a data base. Instructions appeared on the computer screens of each participant, which in addition were read aloud by the person conducting the experiment. After the instructions were read, the experiment commenced, whereby a stimulus word appeared on the computer screen of each participant, who then wrote the first free association word which came to their mind – only one response was possible. Once the participant wrote down their association (or the time ran out for him to write down his association), the next stimulus word appeared on the screen, until the experiment was concluded. The number of stimulus words, as well as their order, was the same for all participants.

As a result, we obtained 260 association lists which consisted of more than 16,000 associated words. Association list derived from the experiment will be used to evaluate algorithm-derived association lists.

2.2.3. *Human association topology*

In this chapter, the associations coming from various sources are compared based on ranked word lists. This does not reflect, however, the complex structure of human associations. These can be represented as weighted graphs with specific words in nodes and associations in the vertices. This graph can be then sub-divided into subnets by starting from a specific stimulus (word) and cutting off the net at a certain distance from this central stimulus. Those subnets can be treated as representative for a specific meaning of a word. The strongest associations are always correlated with the fact that they are bi-directional. But if we look at the each pair of connected words to find what the connection means, we see that connections can differ in meaning, e.g. *home–mother* says that a home is a place specific to a mother in contrast to *home–roof* which says that roof is a part of a building. Having analyzed all the pairs we may find that some of them connect the stimulus word in the same way, e.g. *parents* and *family* connect the *home* on the same principle as *mother*, and *chimney* and *wall* are parts of a building along with the *roof*. This observation shows that the lexical meanings of the stimulus word are organizing subnets in an association network. We show two of them to illustrate the phenomenon. Figure 2.1 shows the sub-net for the meaning *dom* ("home", as a place for family) and Figure 2.2 *dom* ("home", as a building).

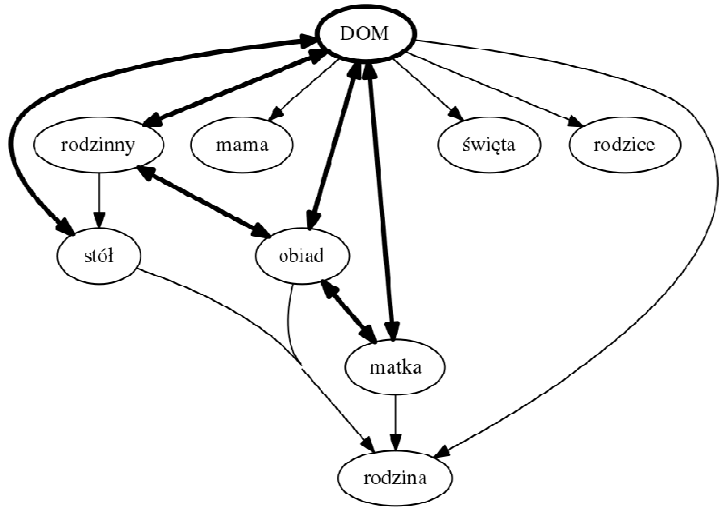


Figure 2.1. *Human association subnet for dom ("home", as a place for family)*

The graph in Figure 2.1 shows the relations between the words: *dom* and *rodzinny* (family; adjective), *stół* (table), *mama* (mum), *matka* (mother), *obiad* (dinner), *święta* (holidays), *rodzice* (parents) and *rodzina* (family).

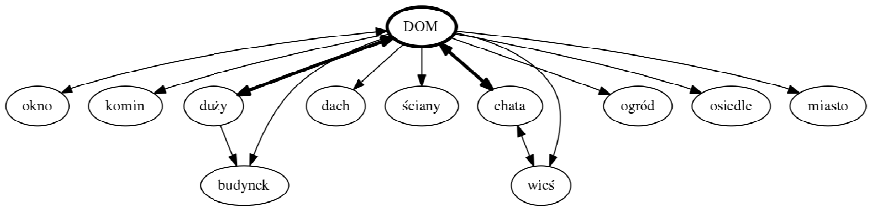


Figure 2.2. *Human association subnet for dom ("house", as a building)*

The graph in Figure 2.2 shows the relations between words: *dom* and *komin* (chimney), *big* (duży), *budynek* (building), *dach* (roof), *ściany* (walls), *chata* (cottage), *wieś* (village), *ogród* (garden), *osiedle* (estate) and *miasto* (city).

Clearly both subnets were identified manually and it is hard to believe that it would be possible to extract those subnets automatically by a use of an algorithm operating solely on the network [GAT 16]. Then, we shall treat all the associations to a particular stimulus as a list disregarding what association means. Then, we may distinguish semantically valid associations comparing the Polish association list with the English associations obtained in the free word association experiment.

2.2.4. *Human associations are comparable*

We shall compare a Polish list derived from our experiment to a semantically equivalent English list derived from the Edinburgh Associative Thesaurus. To illustrate the problem, we selected an ambiguous Polish word *dom*, which refers to the English words *home* and *house*. Those lists will present words associated with their basic stimulus, and ordered in accordance to their strength of association. Due to the varied number of responses (95 for *home* and *house* and 540 for *dom*), we will be using a more qualitative measure of similarity based on the rank of occurring words on them, rather than on a direct comparison of association strength. That list measure $LM_w(l_1,l_2)$, given two word lists l_1 and l_2 and a comparison window,

which will be equivalent to the amount of words matching in l_1 and l_2 in a window of w words taken from the beginning of the lists.

In order to establish some basic expected levels of similarity, we will compare the list obtained in our experiment for the stimulus word *dom*, whose meaning covers both English words *home* and *house*. First, each Polish association word was carefully translated into English, and then the lists automatically looked for identical words. Because words may differ in rank on the compared lists, the table includes the window size needed to match a word on both lists.

dom	home	house
rodzinny (*adv. family*)	house	home
mieszkanie (*flat*)	family	garden
rodzina (*n.family*)	mother	door
spokój (*peace*)	away	boat
ciepło (*warmth*)	life	chimney
ogród (*garden*)	parents	roof
mój (*my*)	help	flat
bezpieczeństwo (*security*)	range	brick
mama (*mother*)	rest	building
pokój (*room*)	stead	bungalow

Table 2.1. *Top 10 elements of the experiment lists for dom (author's experiment) and the EAT lists for home and house*

The lists can be compared separately, but considering the ambiguity of *dom*, we can compare the list of association of *dom* with a list of interspersed (i.e. a list composed of the 1st word related to *home*, next to the 1st word associated with *house*, then the 2nd word related to *home*, etc.) associations of both *home* and *house* lists from the EAT.

w	home + house vs. *dom*	*w*	home vs. *dom*	*w*	*house* vs. *dom*
3	family	3	Family	3	Family
6	garden	9	Mother	6	Flat
9	mother	18	Cottage	6	Garden
12	roof	24	Garden	11	Roof
14	flat	26	Parents	14	Room
18	building	35	Peace	15	Building
19	chimney	41	Security	19	Chimney
26	parents			21	Cottage
30	room			30	Mother
32	brick			32	Brick
35	cottage			34	Security
64	security			40	Warm
65	peace			41	Warmth
74	warm				
75	warmth				

Table 2.2. *Comparison of the experiment list and the EAT lists. Matching words are shown for their corresponding window sizes w for the* $LM_w(l_1,l_2)$ *measure*

The original, i.e. used for comparison human association list, is a list of words associated with a stimulus word ordered by frequency of responses. Unfortunately, we cannot automatically distinguish words which enter into semantic relation to the stimulus word by frequency or by computed association strength, for example in the list associated to the word *table* a semantically unrelated *cloth* is substantially more frequent than *legs* and *leg*, which enter into "part of" relation to the *table* [PAL 64]. The described observation is language independent. The proposed method of comparison truncates from the resulting list language-specific semantic associations, e.g. *home – house* and *house – home* the most frequent on EAT as well as all non-semantic associations, e.g. *home – office* or *house – Jack.* Each resulting list consists of words, each of which is semantically related to a stimulus word. In other words, the comparison of the human association list will automatically extract a sub-list of semantic associations.

2.3. Algorithm efficiency comparison

2.3.1. *The corpora*

In order to compare the association lists with the LSA lists, we have prepared three distinct corpora to train the algorithm. The first consists of 51,574 press notes of the Polish Press Agency and contains over 2,900,000 words. That corpus represents a very broad description of reality, but can be somehow seen as restricted to only a more formal subset of the language. This corpus will be referred to as PAP.

The second corpus is a fragment of the National Corpus of Polish [PRZ 11] with 3,363 separate documents spanning over 860,000 words. That corpus is representative in the terms of the dictionary of the language; however, the texts occurring in it are relatively random, in the sense that they are not thematically grouped or following some deeper semantic structure. This corpus will be referred to as the NCP.

The last corpus is composed of 10 short stories and one novel *Lalka* (The Doll) by Bolesław Prus – a late 19th Century novelist using a modern version of Polish similar to the one used nowadays. The texts are split into 10,346 paragraphs of over 300,000 words. The rationale behind this corpus was to try to model some historically deeply rooted semantic associations with such basic notions as *dom*. This corpus will be referred to in as PRUS.

All corpora were lemmatized using a dictionary-based approach [KOR 12].

2.3.2. *LSA-sourced association lists*

Latent Semantic Analysis is a classical tool for automatically extracting similarities between documents, through dimensionality reduction. A term-document matrix is filled with weights corresponding to the importance of the term in the specific document (term-frequency/inverted document frequency in our case) and then is reduced via Singular Value Decomposition to a lower dimensional space called the concept space.

Formally, the term-document matrix X of dimension $n \times m$ (n terms and m documents) can be decomposed into U and V orthogonal matrices and $\sum$ a diagonal matrix through singular value decomposition:

$$X = U \sum V^T \quad [2.1]$$

This in turn can be represented through a rank k approximation of X in a smaller dimensionally space ($\sum$ becomes a $k \times k$ matrix). We used an arbitrary rank value of 150 in our experiment:

$$X_k = U_k \sum_k V_k^T \quad [2.2]$$

This representation is often used to compare documents in this new space, but as the problem is symmetrical it can be used to compare words. The U_k matrix of dimensions $n \times k$ represents the model of words in the new k-dimensional concept space. We can thus compare the relative similarity of each word by taking the cosine distance between their representations.

The LSA-sourced lists of associations are composed of the ordered list (by cosine distance) from the given word in a model build on each of the tree corpora as described above.

A crucial element in the application of Latent Semantic Analysis [LAN 08] is determining k, the number of concepts that are used to project the data to the reduced k-dimensional concept space. As this parameter is a characteristic of the corpus, and to some degree of the specific application, in this case it has been determined experimentally. For each corpus (PRUS, NCP and PAP), an LSA model has been built for a range of dimensions between 25 and 400 with an increment of 25. For each corpus, the dimension has been chosen as the one that gave the highest sum of matching words from 10 association lists in a window of 1,000 words. The final results, as presented in section 3.4, correspond to a dimension of 75 for PRUS and NCP and 300 for PAP. The calculations were made using the *gensim* topic modeling library.

2.3.3. *LDA-sourced lists*

Latent Dirichlet Allocation is a mechanism used for topic extraction [BLE 03]. It treats documents as probabilistic distribution sets of words or topics. These topics are not strongly defined – as they are identified on the basis of the likelihood of co-occurrences of words contained in them.

In order to obtain ranked lists of words associated with a given word w_n, we take the set of topics generated by LDA, and then for each word contained, we take the sum of the weight of each topic multiplied by the weight of given word w_n in this topic.

Formally, for N topics and w_{ji} denoting the weight of the word *i* in the topic *j*, the ranking weight for the word *i* is computed as follows:

$$w_i = \sum_{j=1..N} w_{ij} * w_{nj} \qquad [2.3]$$

This representation allows us to create a ranked list of words associated with a given word w_n based on their probability of co-occurrence in the documents.

2.3.4. *Association ratio-based lists*

In order to evaluate the quality of the relatively advanced mechanism of Latent Semantic Analysis, we will compare its efficiency to the *association ratio* as presented in [CHU 90], with some minor changes related to the nature of the processed data. For two words *x* and *y*, their association ratio $f_w(x,y)$ will be defined as the number of times *y* follows or precedes *x* in a window of *w* words. The original association ratio was asymmetric, considering only words *y* following the parameter *x*. This approach will, however, fail in the case of texts that are written in languages with no strict word ordering in sentences (Polish in our case) where syntactic information is represented through rich inflection rather than through word ordering. We will use the same value for *w* as is in Church and Hanks [CHU 90] that suggested a value of 5. This measure can be seen as simplistic in comparison with LSA, but, as the results will show, is useful nonetheless.

2.3.5. *List comparison*

First, we have to compare the list obtained automatically from the three corpora for the word *dom* (*home/hose*) with the reference list, i.e. human association list obtained from human subjects in the author's experiment. The comparison will be presented in terms of $LM_w(l_1,l_2)$ for l_1 being the human association list and l_2 being the lists obtained through LSA/LDA similarities and the *association ratio* f_5 as described above. In the comparison we shall apply the three different window sizes to the reference list.

To begin, we shall compare the full human association list that is 151 words long to the lists generated by the algorithms described above. We will arbitrarily restrict the length of automatically generated lists to 1,000 words.

W	PRUS f_5	PAP f_5	NCP f_5	PRUS LSA	PAP LSA	NCP LSA	PRUS LDA	PAP LDA	NCP LDA
10									
25							1		
50	2	1	2				1		1
75	2	4	3	1		1	3		1
100	4	7	9	2		2	4	2	4
150	11	14	17	2	2	2	7	3	6
300	19	24	30	2	6	3	13	8	12
600	34	25	41	4	11	12	22	15	23
1,000	36	43	49	7	13	18	39	23	39

Table 2.3. *$LM_w(l_1,l_2)$ values for different w, for different l_2 from various list sources, l_1 being the human experiment result list*

This may seem excessive, as it contains also a random association of low interest to us – the lists obtained through EAT and the author's list comparison contain only 15 words.

Thus, we will restrict the human association list to only the first 75 words – that was also the length needed to obtain the combined list for *home* and *house* from the EAT.

W	PRUS f_5	PAP f_5	NCP f_5	PRUS LSA	PAP LSA	NCP LSA	PRUS LDA	PAP LDA	NCP LDA
10									
25							1		
50	2		2				1		1
75	2	4	3	1		1	3		1
100	3	5	8	2		1	3		3
150	8	9	10	2	1	1	5		3
300	11	15	21	2	5	1	10	3	8
600	21	23	30	4	7	5	14	8	15
1,000	22	28	33	5	9	6	23	14	22

Table 2.4. *$LM_w(l_1,l_2)$ values for different w, for different l_2 from various list sources, l_1 being the human experiment result list restricted to 75 entries*

As can be seen, automatically generated association lists match some part of the human association list only if we use a large window size. Second, we can observe that the Church–Hanks algorithm seems to generate a list that is more comparable to a human-derived list.

The shorter word list in the EAT (*house*) contains 42 words. The 40 words is the window size, which applied to the author's list, allow us to find all the elements common to the EAT *home/house* combined list and the author's experiment list for *dom*. Therefore, we shall use a 40-word window for comparison.

As we can see this window size seems to be optimal, because it reduces substantially – if compared to the full list – the non-semantic associations for both algorithms.

Finally, we have to test automatically generated lists against the combined human association list, i.e. the list which consists of words, which are present both in the author's list and the EAT lists, presented in Table 2.2.

W	PRUS f_5	PAP f_5	NCP f_5	PRUS LSA	PAP LSA	NCP LSA	PRUS LDA	PAP LDA	NCP LDA
10									
25							1		
50	2		2				1		1
75	2	4	3	1		1	2		1
100	3	5	7	1		1	2		3
150	7	9	9	1		1	4		3
300	8	9	17	1	4	1	8	1	6
600	15	16	22	2	6	5	10	3	11
1,000	16	20	22	3	6	6	16	6	15

Table 2.5. *$LM_w(l_1,l_2)$ values for different w, for different l_2 from various list sources, l_1 being the human experiment result list restricted to first 40 entries*

W	PRUS f_5	PAP f_5	NCP f_5	PRUS LSA	PAP LSA	NCP LSA	PRUS LDA	PAP LDA	NCP LDA
10						1			
25						1	1		
50			1			1	1		1
75		1	3			1	3		1
100		3	3			2	3		3
150	3	4	5			2	5		3
300	4	8	5		1	2	9	2	8
600	8	12	9		2	3	12	7	13
1,000	10	12	12	2	2	3	16	7	14

Table 2.6. *$LM_w(l_1,l_2)$ values for different w, for different l_2 from various list sources, l_1 being the human experiment result list restricted to words that are present in both the author's and the EAT experiment, see Table 2.2*

The results show a tendency similar to that observed during the test of human association list in full length. First, the window size influences the matching number. The second observation is also similar: the list generated by the Church–Hanks algorithm matches the human association list better – it matches 10 or 12 out of 15 words semantically related to the stimulus.

To learn more, we repeated a comparison over a wider range of words. We selected eight words: *chleb* (bread), *choroba* (disease), *światło* (light),

głowa (head), *księżyc* (moon), *ptak* (beard), *woda* (water) and *żolnierz* (soldier). Then, we used the described method to obtain a combined list for the author's experiment and the EAT.

Word	*W*	PRUS f_5	PAP f_5	NCP f_5	PRUS LSA	PAP LSA	NCP LSA	PRUS LDA	PAP LDA	NCP LDA
Bread	25		1			1	1		1	
	100		4	2		1	1	1	2	1
	1,000	1	8	3		2	2	1	3	2
Disease	25		1						1	
	100	1	3	5				1	1	2
	1,000	1	9	8	1	7	2	2	8	8
Light	25	1	1			1		1	1	
	100	3	4	3	1	1		2	2	1
	1,000	3	5	3	4	5	2	5	4	2
Head	25	1		2		1	1		1	1
	100	1	2	4		1	1	1	1	1
	1,000	3	6	6	1	2	3	2	3	2
Moon	25		3	3	1		2	1		1
	100	3	4	5	1		3	1	1	2
	1,000	3	4	6	4	2	5	3	3	7
Bird	25	1	2	1	1		1	1		1
	100	2	4	2	1		2	1	2	3
	1,000	2	5	7	4	3	3	3	2	3
Water	25		1	2	1	1		1	1	1
	100		4	6	2	3	2	3	3	3
	1,000	4	8	10	3	5	6	5	6	5
Soldier	25	2	2	2	2	1	3	2	2	3
	100	2	5	5	2	6	3	2	7	4
	1,000	2	12	9	3	10	4	3	11	5

Table 2.7. *$LM_w(l_1,l_2)$ values for different word stimuli, different w, for different l_2 from various list sources, l_1 being the human experiment result list restricted to entries in both the author's and the EAT experiment*

The table below contains similar comparison, but without restricting the association list to words contained in both experiments.

Word	*W*	PRUS f_5	PAP f_5	NCP f_5	PRUS LSA	PAP LSA	NCP LSA	PRUS LDA	PAP LDA	NCP LDA
Bread	25	1	1	2		1	2		1	1
	100	2	5	6	1	2	5	1	3	4
	1,000	4	19	12	3	4	9	5	6	8
Disease	25		1	1		1			1	
	100	1	3	7		2		1	4	4
	1,000	3	13	14	1	13	8	2	14	7
Light	25	2	1	1	1	1		1	1	
	100	6	6	4	3	1		6	1	1
	1,000	11	15	9	10	9	3	10	9	4
Head	25	3	1	3		3	1		3	1
	100	6	6	7		5	1	1	5	2
	1,000	17	17	12	7	9	7	4	9	7
Moon	25	1	4	6	1		2	1		3
	100	5	5	11	1	1	4	2	1	5
	1,000	5	9	15	7	5	12	6	5	15
Bird	25	1	8	2	2		2	2		2
	100	3	9	5	3	2	2	4	2	3
	1,000	5	13	19	8	9	9	9	9	9
Water	25	1	2	3	1	1	1	1	1	1
	100	3	7	8	2	4	3	3	4	4
	1,000	9	20	21	10	9	15	14	9	17
Soldier	25	1	5	4	1	2	3	1	2	3
	100	2	11	9	4	7	6	6	8	7
	1,000	3	25	22	9	20	11	8	16	10

Table 2.8. *$LM_w(l_1,l_2)$ values for different word stimuli, different w, for different l_2 from various list sources, l_1 being the unrestricted human experiment result list*

As can be seen, the values in the columns corresponding to the f_5 algorithm are clearly better than the corresponding LSA values, regardless of the size of the human lists.

2.4. Conclusion

If we look at our results, we may find that in general they are comparable with the results of related research of Wandmacher [WAN 05] and [WAN 08]. Generally speaking, both the LSA and LDA algorithms generate an association list, which contains only a small fraction of the lexical relationships, which are present in the human association norm. Surprisingly, the Church–Hanks algorithm does much better, which suggests that the problem of how machine-made associations relate to the human association norm should be investigated more carefully. The first suggestion may be derived from [WET 05] – we have to learn more about the relationship between the human association norm and the text to look for a method more appropriate than a simple list comparison. We argue that if a human lexicographer uses contexts retrieved from text by the Church–Hanks algorithm to select those which define lexical meaning, then the list of associations generated by three compared algorithms should be filtered by a procedure that is able to assess the semantic relatedness of two co-occurring words, or we shall look for a new method of co-occurrences selection.

A second suggestion may be derived from an analysis of the human association list. It is well known that such a list consists of responses, which are semantically related to the stimulus, responses which reflect pragmatic dependencies and so-called "clang responses". But within this set of semantically related responses, we can find more frequent direct associations, i.e. such as those which follow a single semantic relation, e.g. "whole – part": *house* – *wall* and not so frequent indirect associations like: *mutton* – *wool* (baranina – rogi), which must be explained by a chain of semantic relations, in our example "source" relationship, i.e. *ram* is a source of *mutton* followed by "whole – part" relation, i.e. *horns* is a part of *ram* or the association: *mutton* – *wool* (baranina – wełna), explained by a "source" relation, i.e. *ram* is a source of *mutton*, followed by "whole – part" *fleece* is a part of *ram*, which is followed by "source" relation, i.e. *fleece* is a source of *wool*. These association chains suggest that some associations are based on a semantic network, which may form the paths explaining indirect associations. Then, it would be very interesting to recognize that human associations may form a network [KIS 73] and test the machine associating mechanism against the association network.

2.5. Bibliography

[BLE 03] BLEI D.M., NG A.Y., JORDAN M.I., "Latent Dirichlet allocation", *Journal of Machine Learning Research*, vol. 3, nos. 4–5, pp. 993–1022, 2003.

[BOR 09] BORGE-HOLTHOEFER J., ARENAS A., "Navigating word association norms to extract semantic information", *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, Groningen, available at: http://csjarchive.cogsci.rpi.edu/Proceedings/2009/papers/621/paper621.pdf, pp. 1–6, 2009.

[BUD 06] BUDANITSKY A., HIRST G., "Evaluating wordnet-based measures of lexical semantic relatedness", *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[CHU 90] CHURCH K.W., HANKS P., "Word association norms, mutual information, and lexicography", *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[DEE 90] DEERWESTER S., DUMAIS S., FURNAS G. *et al*., "Indexing by latent semantic analysis", *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[GAT 14] GATKOWSKA I., "Word associations as a linguistic data", in CHRUSZCZEWSKI P., RICKFORD J., BUCZEK K. *et al*. (eds), *Languages in Contact*, vol. 1, Wrocław, 2014.

[GAT 16] GATKOWSKA I., "Dom w empirycznych sieciach leksykalnych", *Etnolingwistyka*, vol. 28, pp. 117–135, 2016.

[KEN 10] KENT G., ROSANOFF A.J., "A study of association in insanity", *American Journal of Insanity*, vol. 67, pp. 317–390, 1910.

[KIS 73] KISS G.R., ARMSTRONG C., MILROY R. *et al*., "An associative thesaurus of English and its computer analysis", in AITKEN A.J., BAILEY R.W., HAMILTON-SMITH N. (eds), *The Computer and Literary Studies*, Edinburgh University Press, Edinburgh, 1973.

[KOR 12] KORZYCKI M., "A dictionary based stemming mechanism for Polish", in SHARP B., ZOCK M. (eds), *Natural Language Processing and Cognitive Science 2012*, SciTePress, Wrocław, 2012.

[KUR 67] KURCZ I., "Polskie normy powszechności skojarzeń swobodnych na 100 słów z listy Kent-Rosanoffa", *Studia Psychologiczne*, vol. 8, pp.122–255, 1967.

[LAN 08] LANDAUER T.K., DUMAIS S.T., "Latent semantic analysis", *Scholarpedia*, vol. 3, no. 11, pp. 43–56, 2008.

[MOS 96] MOSS H., OLDER L., *Birkbeck Word Association Norms*, Psychology Press, 1996.

[NEL 98] NELSON D.L., MCEVOY C.L., SCHREIBER T.A, *The University of South Florida Word Association, Rhyme, and Word Fragment Norms*, 1998.

[ORT 12] ORTEGA-PACHECO D., ARIAS-TREJO N., BARRON MARTINEZ J.B., "Latent semantic analysis model as a representation of free-association word norms", *11th Mexican International Conference on Artificial Intelligence (MICAI 2012)*, Puebla, pp. 21–25, 2012.

[PAL 64] PALERMO D.S., JENKINS J.J., *Word Associations Norms: Grade School through College*, Minneapolis, 1964.

[POS 70] POSTMAN L.J., KEPPEL G., *Norms of Word Association*, Academic Press, 1970.

[PRZ 11] PRZEPIÓRKOWSKI A., BAŃKO M., GÓRSKI R, *et al*., "National Corpus of Polish", *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznań, pp. 259–263, 2011.

[RAP 02] RAPP R., "The computation of word associations: comparing syntagmatic and paradigmatic approaches", *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, vol. 1, pp. 1–7, 2002.

[RAP 08] RAPP R., "The computation of associative responses to multiword stimuli", *Proceedings of the Workshop on Cognitive Aspects of the Lexicon (COGALEX 2008)*, Manchester, pp. 102–109, 2008.

[SCH 12] SCHULTE IM WALDE S., BORGWALDT S., JAUCH R., "Association norms of German noun compounds", *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Istanbul, available at: http://lrec.elra.info/proceedings/lrec2012/pdf/584_Paper.pdf, pp. 1–8, 2012.

[SIN 04] SINOPALNIKOVA A., SMRZ P., "Word association thesaurus as a resource for extending semantic networks", *Proceedings of the International Conference on Communications in Computing, CIC '04*, Las Vegas, pp. 267–273, 2004.

[WAN 05] WANDMACHER T., "How semantic is latent semantic analysis", *Proceedings of TALN/RECITAL 5,* available at: https://taln.limsi.fr/tome1/P62.pdf, pp. 1–10, 2005.

[WAN 08] WANDMACHER T., OVCHINNIKOVA E., ALEXANDROV T., "Does latent semantic analysis reflect human associations", *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics,* Hamburg, available at: http://www.wordspace.collocations.de/lib/exe/fetch.php/workshop:esslli:esslli_2008_lexicalsemantics.pdf, pp. 63–70, 2008.

[WET 05] WETTLER M., RAPP R., SEDLMEIER P., "Free word associations correspond to contiguities between words in text", *Journal of Quantitative Linguistics*, vol. 12, no. 2, pp. 111–122, 2005.

3

How a Word of a Text Selects the Related Words in a Human Association Network

According to tradition, experimentally obtained human associations are analyzed in themselves, without relation to other linguistic data. In rare cases, human associations are used as the norm to evaluate the performance of algorithms, which generate associations on the basis of text corpora. This chapter will describe a mechanical procedure to investigate how a word embedded in a text context may select associations in an experimentally built human association network. Each association produced in the experiment has a direction from stimulus to response. On the other hand, each association is based on the semantic relation between the two meanings, which has its own direction which is independent from the direction of associations. Therefore, we may treat the network as a directed or an undirected graph. The procedure described in this chapter uses both graph structures to produce a semantically consistent sub-graph. A comparison of the results shows that the procedure operates equally well on both graph structures. This procedure is able to distinguish those words in a text which enter into a direct semantic relationship with the stimulus word used in the experiment employed to create a network, and is able to separate those words of the text which enter into an indirect semantic relationship with the stimulus word.

3.1. Introduction

It is easy to observe that semantic information may occur in human communication, which is not lexically present in a sentence. This phenomenon does not affect the human understanding process, but the performance of a text processing algorithm may suffer from that. Consider, e.g. this exchange: *Auntie, I've got a terrier! – That's really nice, but you'll have to take care of the animal*. The connection between the two sentences

Chapter written by Wiesław LUBASZEWSKI, Izabela GATKOWSKA and Maciej GODNY.

in this exchange suggests that there is a link between *terrier* and *animal* in human memory. A lexical semanticist may explain this phenomenon by the properties of the *hyponymy* relation, which is a transitive one: the pairs, a terrier is a dog, and a dog is an animal, imply that a terrier is an animal [LYO 63, MUR 03]. We can even process this phenomenon automatically using a dictionary such as WordNet. However, there are frequent situations when we require a more complex reasoning to decode the information encoded in a text. Let us take an example: *The survivor regained his composure as he heard a distant barking*. It seems to be obvious that a human reader, who is a native speaker of English, can easily explain the reason behind the change of a survivor's mental state. This person may say for example: *a dog barks* and *a dog lives close (is subsidiary) to a man* and *a man may help the survivor*. However, we can find it impossible to perform such reasoning by an algorithm which uses manually built semantic dictionaries such as WordNet or even FrameNet at their present stage of development [RUP 10].

Then, we can find it reasonable to study the properties of an experimentally built natural dictionary, i.e. an association network that consists of words and naturally preferred semantic connections between them. There exists a reliable method to build such a network. The free word association test [KEN 10], in which the tested person responds with a word associated with a stimulus word provided by a researcher, would provide naturally preferred connections between the stimulus word and the response word. If we perform a multiphase word association test using responses obtained in the initial phase as the stimuli in the next phase of the test, we would create a rich lexical network in which words are linked with multiple links [KIS 73].

Returning to our example, we shall look-up the Edinburgh Associative Thesaurus, which is the first large lexical network built experimentally. We can find here 35 *dog* associations such as, among others:

dog – man, bark, country, pet, gun, collar, leash, lead, whistle

We see the word *dog* directly connected to *bark* and *man*, and that both words co-occur with *collar, lead, leash, pet*, *gun*, which are attributes of the *dog – man* proximity. Then, if we look at the *dog* associations in the Polish lexical network [GAT 14] built in the experiment, where the word *dog* was

not included in the stimuli set, and it is associated only by responses, we can find the following associations:

man, sheep, protector, smoke – dog

As we see in the Polish network, dog is also linked to a man, while other dog-linked words suggest that a dog is working for a man.

Therefore, we may suppose that a study of the meaning connections in the lexical network built by the free association test would provide the data to explain how a word of a text connects in the dictionary and how (if possible) those connections may provide information, which is lexically missing in a text. There are also phenomena observed in the network which may strengthen this supposition. If we look closer at the dog associations, we may find that most of them are directly explicable, e.g. a dog is a pet, a dog has a collar or a dog is a protector. However, there are also on both lists associations which need reasoning to be explained – we call them indirect associations. For example, the association dog – gun in the English network may be explained by the reasoning based on the chain of directly explicable connections: dog is subsidiary to a man and man hunts and man uses a gun. We can find the same situation in Polish for the dog – smoke association: dog is subsidiary to a man and man causes fire and fire produces smoke. Once we identify an indirect association such as dog – gun in the network, we may look in the network for the path which has the dog as a start node and the gun as the end node. If the path is found, we have to assess the path to find if it explains the dog – gun connection. It has been observed that, if a network is rich enough, we may identify more distant associations and explaining paths, e.g. *mutton – horns*, explained by the path *mutton – ram – horns* or the association *mutton – wool*, explained by the path *mutton – ram – fleece – wool*, which were identified manually in the Polish network [GAT 13].

However, before we start looking for the explaining paths in the network, we have to develop a reliable mechanic procedure that takes a word of a text as an input and can find in the network the sub-net (sub-graph) which is optimally related to the word of a text – where optimal means a sub-net in which each node (word) is semantically related to a word of a text. This chapter describes such a procedure.

The procedure to be described was originally designed to operate on the association network treated as an undirected graph [LUB 15]. However, the evaluations of the semantic consistency of the sub-net extracted by the procedure were really encouraging and we therefore decided to expand the procedure to make it able to operate simultaneously on the network treated as a directed graph. The expansion is important because it adapts the procedure to the nature of the network – the network built in the free word association experiment is a directed graph; each connection between two nodes (words) in the network has a direction, always from the stimulus word to the response word. This expansion enables us to really assess a procedure. We shall compare how it operates, on both directed and undirected network structures.

3.2. The network

The network described in this chapter was built via a free word association experiment [GAT 14], in which two sets of stimuli were employed, each in a different phase of the experiment. In the first phase, 62 words taken from the Kent–Rosanoff list were tested as the primary stimuli. In the second phase, the five most frequent responses to each primary stimulus obtained in phase one were used as stimuli. To reduce the amount of manual labor required to evaluate the algorithm output, we used a reduced network, which is based on:

– 43 primary stimuli taken from the Polish version of the Kent–Rosanoff list;

– 126 secondary stimuli which were the three most frequent associations with each primary stimulus.

The average number of associations with a particular stimulus, produced by more than 900 subjects, is approximately 150. Therefore, the total number of stimulus–response pairs obtained for the 168 stimuli, as a result of the experiment, is equal to 25,200. Due to the fact that the analysis of the results produced by an algorithm would require manual work, we reduced the association set through the exclusion of each stimulus–response pair, where the response frequency was equal to 1. As a result, we obtained 6,342 stimulus–response pairs, where 2,169 pairs contain responses to the primary stimuli, i.e. primary associations, and 4,173 pairs that contain responses to the secondary stimuli, i.e. secondary associations. The resulting network consists of 3,185 nodes (words) and 6,155 connections between the nodes.

The experimentally built association network can be described on a graph, where the *graph* is defined as tuple (*V, E*), where *V* is the set of nodes (vertices) and *E* is the set of connections between the two nodes from *V*. The connection between the two nodes may have a *weight*. The experiment result is a list of triples: (*S,A,C*), where *S* is the stimulus, *A* is the association and *C* is the number of participants, who associated *A* with *S*. The *C* represents the association strength, which can be converted into a connection weight of C_w, counted as follows: $C_w = S_c/C$, where S_c is the total of all responses given to the stimulus *S*. Then, we may treat the association network as a *weighted* graph, which is a tuple (*V, E, w*), where *w* is the function that assigns a weight to every connection.

As each stimulus–association (response) pair has a direction, always from the stimulus to the response, we may consider an association network as a *directed* graph [KIS 73], which means that each connection between the two nodes (*v1, v2*) has a direction, i.e. it starts in *v1* and ends in *v2* – this kind of connection is called an *arc*. On the other hand, if we recognize that the connection (*v1, v2*) is a semantic relation between the meanings of the two words, then we have to recognize that the stimulus–response direction and the direction of semantic relations between the two meanings may differ. Let us consider the associations: *chair – leg* and *leg – chair*. In both cases, the associated meanings are connected by the same semantic relation, i.e. *meronymy* [MUR 03], which has a direction from a part, e.g. *leg* to a whole *chair*. The same phenomenon may be observed with respect to the *hyponymy* relation, which goes from a subordinate *terrier* to a superordinate *dog* meaning and the direction of the relation does not depend on the direction of the association *terrier – dog* or *dog – terrier*. Therefore, we can treat the association network as an *undirected* graph, which means that the connection between the two nodes (*v1, v2*) has no direction, i.e. (*v1,v2*) = (*v2, v1*).

The *path* in the graph is a sequence of nodes that are connected by edges or arcs. The path *length* is the number of nodes along the path. Path *weight* is the sum of the weights of the connections in the path. The *shortest* path between two nodes (*v1, v2*) is the path where the path weight is smaller than the weight of the direct connection between *v1* and *v2*.

3.3. The network extraction driven by a text-based stimulus

If both the network and the text are structures built from words, then we may look for an efficient algorithm that can identify in the text the stimulus word used in an experiment performed to build a network and a reasonable number of direct associations with this stimulus. Words identified in the text may serve as the starting point to extract a sub-graph from the network, which will contain as many associations as possible. The semantic relationship between the nodes of a returned sub-graph will be the subject of an evaluation.

In more technical language, the algorithm should take a graph (association network) and the subset of its nodes identified in a text (*extracting nodes*) as input. Then, the algorithm creates a sub-graph with all extracting nodes as an initial node set. After that, all the connections between the extracting nodes which exist in the network are added to the resulting sub-graph – these connections are said to be *direct*. Finally, every direct connection is checked in the network to determine whether it can be replaced with a shortest path, i.e. a path which has a path weight lower than the weight of the direct connection and a node number less than or equal to the predefined path length. If such a path is found, it is added to the sub-graph – which means adding all the path's nodes and connections. If we apply this procedure to each text of a large text collection, and if we merge the resulting text sub-graphs, we may evaluate the sub-graph created for a particular stimulus word.

3.3.1. *Sub-graph extraction algorithm*

The source graph G, extracting nodes EN and maximum number of intermediate nodes in path l are given. First, an empty sub-graph, SG, is created, and all extracting nodes, EN, are added to a set of nodes (vertices) V_{sg}. In the next set of steps, the ENP of all pairs between the nodes in the EN is created. For every pair in the ENP the algorithm checks if the connection between the paired nodes $v1$, $v2$ exists in G. If it does, this connection is added to the sub-graph SG set of connections E_{sg}. Then, the shortest path sp between $v1$ and $v2$ is checked in G. If the shortest path sp is found, i.e. the sp weight is lower than the weight of the direct connection ($v1$, $v2$) and the number of the shortest path intermediate nodes is less than l ($length(sp) - 2$, (–2 because the start and end nodes are not intermediate), then the sp path is

added to the sub-graph *SG* by adding its nodes and connections to the appropriate sets V_{sg} and E_{sg}. Finally, the sub-graph *SG* is returned.

NEA (G, EN, l)
Input: G – (V_g, E_g, w_g) – graph (set of nodes, set of connections, weighting function), EN - extracting nodes, l - maximal number of intermediate nodes in the path
Output: SG - (V_{sg}, E_{sg}, w_{sg}) - extracted sub-graph

```
1  Vsg ← Vg;
2  Esg ←Ø;
3  wsg ←wg;
4  ENP ←pairs(EN );
5  for each v1 , v2 ϵ ENP do
6        if conns(v1, v2) ϵ G then
7              Esg ← Esg+ conns(v1, v2) ;
8              sp ← shortest_path(v1, v2);
9              if weight(sp) < wg(conns(v1, v2)) and length(sp) - 2 <= l then
10                   for each v ϵ nodes(sp) do
11                         if not v ϵ Vsg then
12                               Vsg ←Vsg + v;
13                         end
14                   end
10                   for each e ϵ conns(sp) do
11                         if not e ϵ Esg then
12                               Esg ←Esg + e;
13                         end
14                   end
17             end
18       end
19 end
   return SG
```

It seems to be clear that the size of the sub-graph created by the algorithm depends on the number of extracting nodes given at the input. As the texts may differ in the number of primary associations with a particular stimulus which would serve as extracting nodes, there is a need for a procedure that controls the number of extracting nodes used by the network extracting algorithm.

3.3.2. *The control procedure*

The procedure controls the number of extracting nodes *EN* and the sub-graph *SG* size. In order for it to be used to build a sub-graph for a given stimulus, the text must contain a stimulus *S* and at least *dAn* direct associations with the stimulus. The $dAn = 2$ is chosen as a starting value for the extracting algorithm, which means that if the text has $dAn < 2$, the text is omitted. If the text has $dAn \geq 2$, the text is used for sub-graph extraction. First, the stimulus and the $dAn = 2$ primary associations are passed as extracting nodes to the network extracting algorithm *NEA*. Then, the number of nodes in the returned sub-graph is counted. In the next step, the *dAn* is incremented by 1 and the new set of extracting nodes is passed to the *NEA*. The returned sub-graph size is evaluated, i.e. the number of nodes of the sub-graph based on $dAn + 1$ is multiplied by the sub-graph size control parameter *Ss*, which tells us which fraction of the base sub-graph, i.e. the sub-graph created for a start value of $dAn = 2$, must exist in the sub-graph created for $dAn + 1$. For example, $Ss = 0.5$ means that at least half of the nodes from the base sub-graph must remain in the sub-graph created after the incrementing of the *dAn*. If the newly created sub-graph does not match the condition set by *Ss*, the procedure stops and the sub-graph created in the previous step becomes the final for a particular text. If the newly created sub-graph matches the condition set by *Ss*, the *dAn* is incremented by 1, and a new sub-graph is created.

3.3.3. *The shortest path extraction*

Figures 3.1 and 3.2 represent a subset of the experimental network, treated respectively as directed and undirected graphs. Each graph consists of such nodes as: *chleb* “bread”, *masło* “butter”, *jedzenie* “food”, *ser* “cheese”, *mleko* “milk”, *dobry* “good”, *kanapka* “sandwich” and *żółty* “yellow” linked by connections produced by the free word association experiment.

Figure 3.1 represents the concept of normalizing the directed network, if a path shorter than the one directly linking two nodes can be found. “Shorter” in this case means that the sum of weights of the path connections is smaller than the weight of the direct connection. On this specific example, the dotted connection between the nodes is replaces the original black one. It is caused by the fact that the path *ser* →*jedzenie* → *chleb* → *masło* has a weight sum of 84, which is lower than the direct connection weight of 200 for the nodes *masło* → *ser*.

The same reasoning applies to the experimental net being represented by an undirected weighted graph (Figure 3.2).

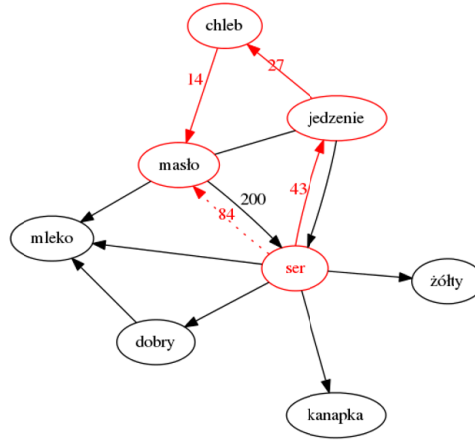


Figure 3.1. *Shortest path for a directed network. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

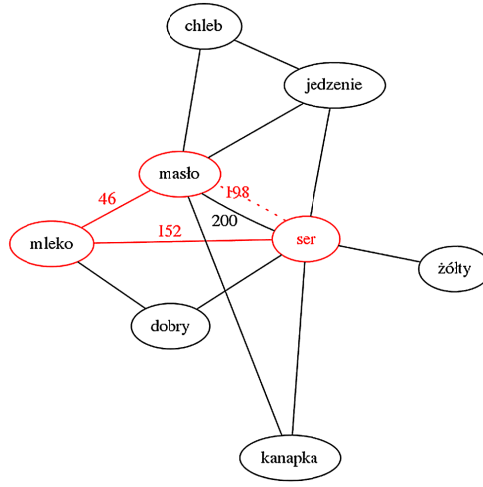


Figure 3.2. *Shortest path for an undirected network. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

In the case of the undirected graph, we treat it just as a directed graph with symmetrical connections between nodes, i.e. $(v1,v2) = (v2, v1)$. We can

see from Figure 3.2 that the connection *ser – masło* is replaced by the same path *ser –jedzenie – chleb – masło* as in the directed graph, and that another shortest path for the *ser – masło* connection is found, i.e. the path *masło – mleko – ser*, with a path weight of 198 which is smaller than 200, which is the weight of the *ser – masło* direct connection.

In both cases, the Dijkstra's classic shortest path algorithm has been applied. However, the sub-graph extracting algorithm *NEA* will reject any shortest path which does not meet the condition set by the *l* parameter.

3.3.4. *A corpus-based sub-graph*

First, a separate sub-graph for each primary stimulus was created for each text in the corpus. All sub-graphs were obtained with empirically adjusted parameters [HAR 14], such as: intermediate nodes in the path of $l = 3$ for the extracting algorithm, and direct associations with the stimulus minimum $dAn = 2$, with a sub-graph adjusting parameter of $Ss = 0.5$ for the control procedure. Then, the text-based sub-graphs obtained for a specific primary stimulus were merged into the corpus-based primary stimulus sub-graph, i.e. all sets of nodes and all sets of edges were merged, forming a multiple set union. Finally, the corpus-based primary stimulus sub-graph was trimmed, which means that each non-connected node was removed from the final sub-graph, and each open path (paths with which the end node had not connected) which had more than two edges between the stimulus and the end node was reduced to match the network-forming principle that a stimulus (A) produces an association (B), which then serves as a stimulus to produce an association (C). Afterwards the reduced path takes the form A – B – C.

3.4. Tests of the network extracting procedure

3.4.1. *The corpus to perform tests*

To test the original procedure, we employed the three stylistically and thematically different corpora, i.e. the PAP corpus which consists of 51,574 press releases of the Polish Press Agency, and contains over 2,900,000 words, the sub-corpus of the National Corpus of Polish with a size of 3,363 separate documents spanning over 860,000 words, and a literary text corpus which consists of 10 short stories and the novel *Lalka* (The Doll) written by the influential novelist, Bolesław Prus. All three corpora were lemmatized

using a dictionary-based approach [KOR 12]. The procedure performed equally well on all three corpora. Then, we decided to perform the test described below on the largest corpus, which is that of the PAP.

3.4.2. *Evaluation of the extracted sub-graph*

To evaluate the quality of the extracted sub-graph, we shall use two separate evaluation criteria: first, to test the semantic consistency of the sub-graph and second, to test how the sub-graph matches the text collection.

3.4.2.1. *Semantic consistency of the sub-graph*

To perform the evaluation, each of the 6,342 stimulus–response pairs used to build the network were manually evaluated. The evaluation was necessary because of the observation that the free word association experiment may produce so-called clang associations, i.e. words that sound like a stimulus or rhyme with the stimulus, e.g. *house – mouse* and the idiom completion associations, e.g. *white – house*, which forms a multipart lexical unit and, therefore, does not reflect the meaning relation between stimulus and response [CLA 70]. We had expanded this observation treating all associations which introduce proper names, e.g. *river – Thames* and the not so frequent deictic association, e.g. *girl – me* as non-semantic.

The evaluation is as follows. If the stimulus is semantically related to the response as in *dom – ściana* (house – wall), the pair is marked as semantic, otherwise the pair is considered to be a non-semantic one, e.g. *góra – Tatry* (mountain – the proper name) or *dom – mój* (house – my).

Then, the sub-graph nodes were evaluated consecutively along a path in the following way. If the two connected nodes matched a stimulus–response pair marked as semantic, then the right node was marked as semantic – *Sn*. If the two connected nodes matched a non-semantic stimulus–response pair, then the right node was marked as non-semantic one – *nSn*. If the two connected nodes did not match any stimulus–response pair, then both nodes were marked as *nSn*, except the stimulus node which is in principle a semantic one. After the final pair of a path was evaluated, the evaluation of the connection of the start node (stimulus) and the end node of the path was evaluated to check the semantic consistency of the path. As a result, a

non-semantic node *nSn* is considered to be any end node (association) that does not have a semantic relation to the start node (stimulus), even if it has a semantic relation with a preceding node, e.g. the path: *krzesło – stół – szwedzki* (chair – table – Swedish), where pairs *krzesło – stół* and *stół – szwedzki* enter into a semantic relation, but the stimulus *krzesło* "chair" does not enter into a semantic relationship with the association *szwedzki* "Swedish".

3.4.2.2. *Matching the sub-graph and a text collection*

To assess how the extracted sub-graph relates to a text collection, we have to match each text that contains a particular stimulus against the sub-graph extracted for this stimulus. Then, we have to count the network nodes (words) that were recognized both in the texts and in the sub-graph *SnT*. After that, we have to match the entire set of direct associations with a particular stimulus which is present in the network against the texts. This is performed in order to recognize network nodes (words) which are present in the network but were rejected by the algorithm, and therefore, are not present in the sub-graph *TnS*.

3.4.3. *Directed and undirected sub-graph extraction: the comparison*

Now, we can present the results for each primary stimulus, where the sub-graph for each primary stimulus word is evaluated. To compare directed and undirected sub-graphs extracted for each stimulus, we shall use all data obtained in the sub-graph evaluation process, i.e.:

– *Sn*: the number of nodes in the sub-graph created by the algorithm;

– *nSn*: the number of non-semantic nodes in the sub-graph recognized by a sub-graph evaluation;

– *SnT*: the number of network nodes (words) which were recognized both in the texts and in the sub-graph;

– *TnS*: the number of network nodes (words) which are present in the texts but were rejected by the algorithm, and therefore are not present in the sub-graph.

Before we start the evaluation per stimulus, we have to show the joint results of the evaluation of 43 stimuli. To perform this analysis, we must determine the total number of nodes in the network – *Nn*. Table 3.1 shows the joint result for all sub-graphs based on the PAP corpus.

Stimuli	Nn	Sn	nSn	SnT	TnS	Graph
43	3,185	898	65	710	38	undirected
43	3,185	878	64	788	64	directed

Table 3.1. *Joint evaluation of 43 stimuli*

If we look at Table 3.1 and compare the number of network nodes *Nn* and the sum of *SnT* (network nodes retrieved in the text to extract a sub-graph) and *TnS* (network nodes present in text but rejected by an algorithm), we can discern that only a fraction of the nodes (words) which are present in the network appear in the large text collection – 0.234 for the undirected network and 0.267 for the directed one. This score is substantially lower than the sub-graph node *Sn*-to-network node *Nn* ratio, which is 0.281 for an undirected network and 0.275 for a directed one. It can be said that these figures show the relation between the language dictionary (the network) and the use of a dictionary to produce texts. The *nSn* value (non-semantic nodes in the sub-graph) shows that the non-semantic nodes in sub-graphs are only 0.072 of the total sub-graph nodes, in both undirected and directed networks. This result shows both the semantic consistency of the empirically built association network, and the quality of the cautious method for building a sub-graph described in this chapter.

Finally, the difference in size of *Sn, SnT* and *TnS* may reflect the differences between the directed and undirected graph structures, which influence the use of the words of a text to extract a sub-graph. We shall provide a detailed analysis later on.

3.4.4. *Results per stimulus*

A more detailed evaluation of the results will be possible if we look at the results obtained for each particular primary stimulus. These results are shown in Table 3.2.

	Directed network				Undirected network			
Stimulus	**Sn**	**nSn**	**SnT**	**TnS**	**Sn**	**nSn**	**SnT**	**TnS**
baranina “mutton”	3	0	3	0	5	0	4	0
chata “cottage”	7	1	6	0	10	0	8	0
chleb “bread”	17	2	16	4	17	0	15	0
chłopiec “boy”	19	1	19	0	22	0	13	1
choroba “illness”	41	2	30	1	30	1	22	0
doktor “doctor”	19	2	17	2	24	0	13	0
dom “house/home”	51	1	51	4	35	1	32	0
dywan “carpet”	3	0	3	0	7	1	5	0
dziecko “child”	51	7	46	2	27	5	26	0
dziewczyna “girl”	14	2	12	0	9	0	7	0
głowa “head”	28	1	28	0	27	3	30	0
góra “mountain”	28	2	26	0	20	4	11	1
jedzenie “food”	27	4	23	1	22	2	13	0
kapusta “cabbage”	13	2	10	2	11	2	6	0
król “king”	14	1	14	2	15	0	13	1
krzesło “chair”	6	2	4	1	7	2	3	0
księżyc “moon”	15	0	15	4	18	0	15	0
lampa “lamp”	5	0	5	1	4	0	4	0
łóżko “bed”	11	2	7	2	7	1	7	1
mężczyzna “man”	33	2	33	6	61	5	38	0
miasto “city”	32	2	32	0	37	3	31	2
mięso “meat”	33	3	28	3	36	1	28	1
motyl “butterfly”	3	0	3	1	12	1	8	0
obawa “fear”	12	1	10	0	10	1	6	0
ocean “ocean”	9	0	8	0	17	1	16	0
okno “window”	22	5	16	1	25	3	16	0
orzeł “eagle”	15	1	15	0	14	1	10	0
owca “sheep”	15	2	11	4	19	2	19	3
owoc “fruit”	27	2	26	0	30	3	15	1
pająk “spider”	3	0	3	0	5	1	4	0
pamięć “memory”	24	1	20	2	22	0	32	3
podłoga “floor”	14	3	11	1	8	1	4	0
praca “work”	54	2	48	2	56	6	44	3
ptak “bird”	20	0	20	1	23	1	19	1
radość “joy”	19	1	16	1	25	0	26	9

ręka “hand”	33	5	29	0	10	1	12	1
rzeka “river”	24	1	19	2	20	0	19	2
ser “cheese”	8	0	7	1	10	0	8	0
sól “salt”	9	0	9	0	17	2	10	0
światło “light”	16	0	14	2	12	2	11	3
woda “water”	48	0	42	4	68	8	43	4
wódka “vodka”	10	1	10	4	10	0	10	0
żołnierz “soldier”	23	0	23	3	34	0	34	1

Table 3.2. *Evaluation for each primary stimulus word*

The joint evaluation has shown that the procedure which operates on an undirected network produces a slightly larger sub-graph. However, if we look at the differences per stimulus in Figure 3.3, which compare the sub-graph size of the directed network with the undirected one, we may find that any differences appear to be stimulus dependent. Figure 3.3 shows that *Sn* size rises simultaneously for both networks and only *Sn* for *dziecko* “child” (+24), *ręka* “hand” (+23), *dom* “home/house” (+16), *choroba* “illness” (+11), *żołnierz* “soldier” (–11), *woda* “water” (–20) and *mężczyzna* “man” (–28) may reflect a difference in network structures. We have to add that the listed words do not share substantial semantic features.

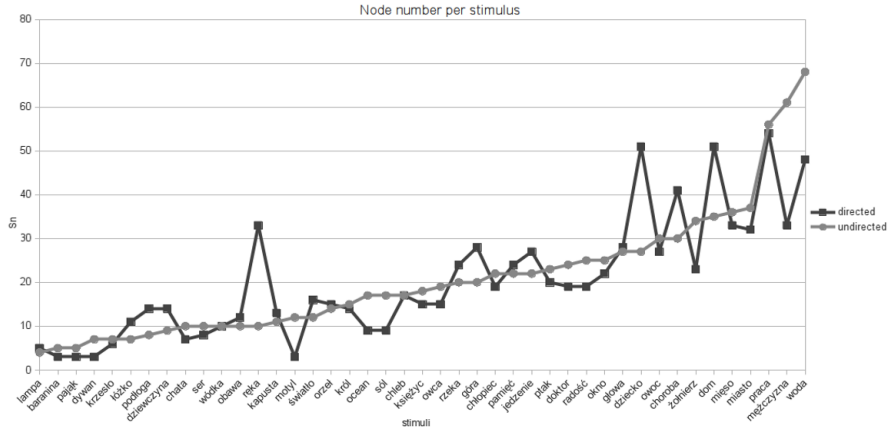


Figure 3.3. *Sub-graph node number per stimulus*

Having compared the sub-graph size, we can analyze the *nSn* – negative nodes in the sub-graph. This can be seen in Figure 3.4, which shows the *nSn*-to-*Sn* ratio per stimulus; the stimuli are ordered by the sub-graph size. We can see that only 17 out of 43 sub-graphs extracted from an undirected network do not contain a non-semantic node, while for a directed network, it is only 13. It is interesting to observe that only five stimuli words, i.e. *baranina* "mutton", *księżyc* "moon", *lampa* "lamp", *ser* "cheese" and *żołnierz* "soldier" share this property in both network structures. The differences in the *nSn*-to-*Sn* ratio seem to be network structure dependent.

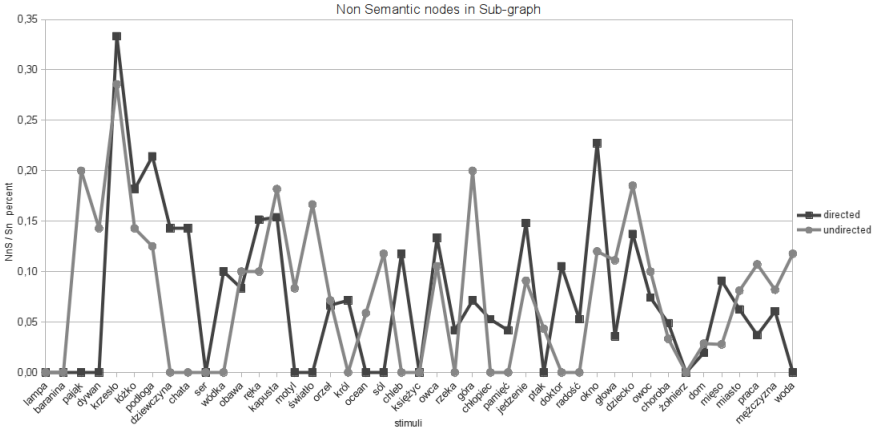


Figure 3.4. *Non-semantic nodes in the sub-graph*

At first glance, we may say that the stimuli status with respect to *SnT* and *TnS* seems to be similar for both directed and undirected networks. The differences in the size of the *SnT* (sub-graph nodes retrieved in text) may be observed in Table 3.2, which seem to be random and corpus dependent – e.g. the stimulus word *dywan* "carpet" occurred in only seven texts and only two of them were rich enough to provide extracting nodes (the stimulus word and two direct associations). The use of *SnT* words to create a sub-graph may depend on the directed or undirected network structure; however, we cannot prove this without separate research.

Finally, we have to analyze the *TnS*, i.e. associations which are present both in the network and texts but are not present in the sub-graph, as the algorithm had rejected them. First, we may observe that the algorithm operating on a directed network rejects more text occurring nodes, which we

may correlate with the smaller number of sub-graph nodes for a directed network. The second observation is that there are only 10 stimuli which have the rejected text occurring nodes for both directed and undirected networks. It seems to be reasonable to look at these rejected network nodes. The full list of rejected nodes for all 10 stimuli is shown in Table 3.3. To save space, we shall use only the English translation of the rejected nodes.

Stimulus	***TnS* – undirected network**	***TnS* – directed network**
król "king"	law	kingdom, scepter
mięso "meat"	breakfast	pork, beef, cow
owca "sheep"	*horns, meat, baby sheep	meadow, mountain, skin, wool
pamięć "memory"	work, *mark, *will	limited, permanent
praca "work"	machine, *decision, hand	relax, difficult
radość "joy"	heartfelt, fear, to enjoy, disappointment, despair, big, disaster, worry, joyful	oda
rzeka "river"	*air, *earth	stream, *forest
światło "light"	lightness, reading, room	darkness, white
woda "water"	depth, salt, *sand, wave	thirst, *desert, drink, wet
żołnierz "soldier"	military	bravery, drill, *sea

Table 3.3. *Rejected nodes for 10 stimuli*

As we look at the nodes rejected by the algorithm operating on both networks, we find that all the words are semantically related to a stimulus, and for most of them, we can directly explain the connection between the stimulus and the association, e.g. *king* "makes/executes" a *law* for an undirected network and *king* "possesses" a *kingdom* and *scepter* is *king's* "attribute". However, some of those rejected nodes (marked by the asterisk) do not relate directly to the stimulus, e.g. *sheep – horns, water – desert*, and we can explain them by a sequence of direct connections, i.e. *sheep – ram – horns* and *water – thirst – search – desert*. That is to say that all words marked by an asterisk relate to a stimulus in the same way as indirect associations. Therefore, we can say that the method described in this chapter may help to identify indirect associations, which are present in the network. It is much easier to manually check a short list of nodes rejected by the algorithm, and then to manually check the entire network. Once the indirect associations are identified, we may rather easily construct an automatic

procedure which would search for the paths explaining those indirect associations.

3.5. A brief discussion of the results and the related work

The proposed method for a text-driven extraction of an association network is simple and cautious on graph operations. The quality of sub-graphs extracted for stimuli words such as *pająk* "spider", *lampa* "lamp" and *dywan* "carpet", which occurred in a really small number of texts, seems to prove that the extracting algorithm does not depend on the number of texts used for network extracting. If it is true, the algorithm may serve as a reliable tool for extracting an association network on the basis of a single text, which may provide data to study how a particular direct association retrieved in a text may influence the sub-graph size and content. That is to say that we may observe how the sub-graph of lamp (Figure 3.5) may change if a text replaces the direct association *ulica* "street" with the direct association *krzesło* "chair", which has its own sub-graph created separately as shown in Figure 3.6.

The *lampa* sub-graph consists of directly associated nodes provided by the text *ulica* "street", *żarówka* "light bulb", *światło* "light" and the connection *żarówka – światło* added by the algorithm.

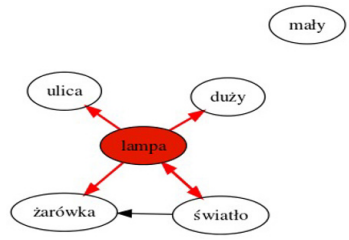


Figure 3.5. *Sub-graph lampa "Lamp". For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

The *krzesło* sub-graph consists of directly associated nodes: *stół* 'table', *dom* 'home', *stary* 'old' provided by the text and *obiad* 'dinner', *rodzinny* 'family' added by the text and *obiad* 'dinner', *rodzinny* 'family' added by the algorithm.

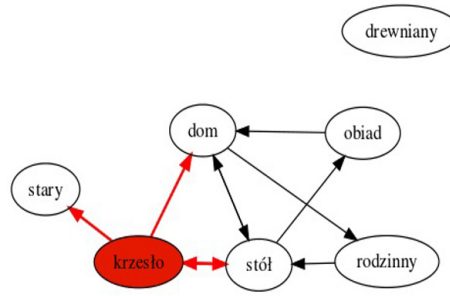


Figure 3.6. *Sub-graph krzesło "chair". For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

The study of a single text seems to be justified because a human reader comprehends just the text, not a text collection. It would be interesting to compare the graph of a text extracted word by word using our method with the graph of a text built solely on the basis of a text collection (e.g. [LOP 07, WU 11, AGG 13]). This should be the subject of further investigation.

The analysis of words which were rejected by the algorithm in the sub-graph extraction process suggests that the text-driven network extracting procedure may serve as a tool to provide the data which would locate the indirect association in a large network. This is a task which would be extremely hard to do manually. Having identified the indirect association, we may search the network automatically to find all paths that would explain those indirect associations. These explanatory paths may bring the new data to the study of the human association mechanism as analyzed by Clark [CLA 70].

However, if we look at an association network from the perspective of a computer program aimed to emulate human reasoning, we will find that an experimentally obtained connection between the two words does not provide explicit information as to what that connection means. However, it seems to be clear that only explicit information on the manner in which a *dog* relates to a *man* may serve as a basis for reasoning about a *survivor's* mental state in the introductory example. This means that we have to classify connections between words to make a network usable for computer programs to perform human-like reasoning. We shall not discuss the possible classification methods (e.g. [DED 08, GAT 15]), but we have to stress that the proper classification must recognize that the connection between two nodes in the

lexical network reflects features that organize a particular type, e.g. *dog, flower*, or particular aggregate, e.g. *furniture, water* structures in the network [SOW 00].

Finally, we have to realize why an experimentally built association network does not entirely match a network built automatically from a text collection. Since an influential study by Rapp [RAP 02], an experimentally built association network served as a norm to evaluate associations generated by different statistical algorithms that operate solely on a text collection (e.g. [WAN 08, GAT 13, UHR 13]). We have to agree that text-generated associations reflect text contiguities [WET 05]. However, we may add that, contrary to text-derived associations, the associations obtained by the free word association experiment represent features that define a lexical meaning. If we compare the results of the *Wortschatz* algorithm operating on the Polish newspaper text collection [BIE 07], we can find that the word *dom* (home/house) is associated with many different verbs, e.g. *kupić* (buy), *uderzyć* (hit), *wybudować* (build), *spłonąć* (burn), *stoi* (standing), *wjechał* (struck), *zniszczyć* (destroy), and *mieć* (possess), which may be associated with many different objects, while, in the experimental network described in this paper, *dom* is associated with a single verb *mieszkać* "to dwell" and this particular verb is specific to *dom*, because *to dwell* defines the destination of the object and the place called *dom*.

3.6. Bibliography

[AGG 13] AGGARWAL C.C., ZHAO P., "Towards graphical models for text processing", *Knowledge and Information Systems*, vol. 36, no. 1, pp. 1–21, 2013.

[BIE 07] BIEMANN C., HEYRER G., QUASTHOFF U. *et al.*, "The Leipzig Corpora Collection – Monolingual corpora of standard size", *Proceedings of Corpus Linguistics 2007*, Birmingham, pp. 1–12, 2007.

[CLA 70] CLARK H.H., "Word associations and linguistic theory", in LYONS J. (ed.), *New Horizons in Linguistics*, Penguin Books, Harmondsworth, 1970.

[DED 08] DE DEYNE S., STORMS G., "Word associations: network and semantic properties", *Behavior Research Methods*, vol. 40, no. 1, pp. 213–231, 2008.

[GAT 13] GATKOWSKA I., KORZYCKI M., LUBASZEWSKI W., "Can human association norm evaluate latent semantic analysis", in SHARP B., ZOCK M. (eds), *Proceedings of the 10th International Workshop on Natural Language Processing and Cognitive Science 2013*, Marseille, 2013.

[GAT 14] GATKOWSKA I., "Word associations as a linguistic data", in CHRUSZCZEWSKI P., RICKFORD J., BUCZEK K. *et al*. (eds), *Languages in Contact*, vol. 1, Wrocław, 2014.

[GAT 15] GATKOWSKA I., "Empiryczna sieć powiązań leksykalnych", *Polonica*, vol. 35, pp. 155–178, 2015.

[HAR 14] HARĘZA M., Automatic text classification with use of empirical association network, Master Thesis, AGH, University of Science and Technology, Kraków, 2014.

[KEN 10] KENT G., ROSANOFF A.J., "A study of association in insanity", *American Journal of Insanity*, vol. 67, no. 2, pp. 317–390, 1910.

[KIS 73] KISS G.R., ARMSTRONG C., MILROY R. *et al.*, "An associative thesaurus of English and its computer analysis", in AITKEN A.J., BAILEY R.W., HAMILTON-SMITH N. (eds), *The Computer and Literary Studies*, Edinburgh University Press, 1973.

[KOR 12] KORZYCKI M., "A dictionary based stemming mechanism for Polish", in SHARP B., ZOCK M. (eds), *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science 2012*, Wrocław, 2012.

[LOP 07] LOPES A.A., PINHO R., PAULOVICH F.V. *et al.*, "Visual text mining using association rules", *Computers and Graphics*, vol. 31, pp. 316–326, 2007.

[LUB 15] LUBASZEWSKI W., GATKOWSKA I., HARĘZA M., "Human association network and text collection", in SHARP B., DELMONTE R. (eds), *Proceedings of the 12th International Workshop on Natural Language Processing and Cognitive Science 2015*, De Gruyter, Berlin, 2015.

[LYO 63] LYONS J., *Structural Semantics. An Analysis of Part of the Vocabulary of Plato*, Blackwell, Oxford, 1963.

[MUR 03] MURPHY M.L., *Semantic Relations and the Lexicon*, Cambridge University Press, Cambridge, 2003.

[RAP 02] RAPP R., "The computation of word associations: comparing syntagmatic and paradigmatic approaches", *Proceedings of the 19th International Conference on Computational Linguistics*, vol. 1, Taipei, pp. 1–7, 2002.

[RUP 10] RUPPENHOFER J., ELLSWORTH M., PETRUCK M.R.L. *et al.*, *FrameNet II: Extended Theory and Practice*, Berkeley University, 2010.

[SOW 00] SOWA J.F., *Knowledge Representation. Logical, Philosophical, and Computational Foundations*, vol. 13, Brooks/Cole, Pacific Grove, 2000.

[UHR 13] UHR P., KLAHOLD A., FATHI M., "Imitation of the human ability of word association", *International Journal of Soft Computing and Software Engineering*, vol. 3, no. 3, pp. 248–254, 2013.

[WAN 08] WANDMACHER T., OVCHINNIKOVA E., ALEXANDROV T., "Does latent semantic analysis reflect human associations", *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, Hamburg, 2008.

[WET 05] WETTLER M., RAPP R., SEDLMEIER P., "Free word associations correspond to contiguities between words in text", *Journal of Quantitative Linguistics*, vol. 12, no. 2, pp. 111–122, 2005.

[WU 11] WU J., XUAN Z., PAN D., "Enhancing text representation for classification tasks with semantic graph structures", *ICIC International*, vol. 7, no. 5(b), pp. 2689–2698, 2011.

4

The Reverse Association Task

Free word associations are the words human subjects spontaneously come up with upon presentation of a stimulus word. In experiments comprising thousands of test subjects, large collections of associative responses have been compiled. In previous publications, it was shown that these human associations can be resembled by statistically analyzing the co-occurrences of words in large text corpora. In this chapter, we consider the reverse question, namely whether the stimulus can be predicted from the responses. We call this reverse association task and present an algorithm for approaching it. We also collected human data on the reverse association task, and compared them with the machine-generated results.

4.1. Introduction

Word associations have always played an important role in psychological learning theory, and have been investigated not only in theory, but also in experimental work where, for example, such associations were collected from human subjects. Typically, the subjects are given questionnaires with lists of stimulus words, and were asked to write down for each stimulus word the spontaneous association which first came to mind. This led to collections of associations, the so-called association norms, as exemplified in Table 4.1. Among the best known association norms are the Edinburgh Associative Thesaurus (EAT) [KIS 73], the Minnesota Word Association Norms [JEN 70, PAL 64] and the University of South Florida Free

Chapter written by Reinhard RAPP.

Association Norms [NEL 98]. More recently, attempts have been made to use crowd sourcing methods for collecting associations in various languages (*Jeux de mots*[1] and *Word Association Study*[2]). In this way, researchers are able to collect much larger datasets than was previously possible.

Association theory, which can be traced back to Aristotle in ancient Greece, has often stated that our associations are governed by our experiences. For example, more than a century ago, William James [JAM 90] formulated this in his book, *The Principles of Psychology*, as follows:

> "Objects once experienced together tend to become associated in the imagination, so that when any one of them is thought of, the others are likely to be thought of also, in the same order of sequence or coexistence as before. This statement we may name the law of mental association by contiguity."

This citation is talking of *objects*, but the question arose whether for words the same principles might apply, and with the advent of corpus linguistics, it was possible to verify this experimentally by looking at the distribution of words in texts. Among the first to do so were [CHU 90], [SCH 89] and [WET 89].

Their underlying assumption was that in text corpora, strongly associated words often occur in close proximity. This is actually confirmed by corpus evidence: Figure 4.1 assigns to each stimulus word position 0, and displays the occurrence frequencies of its primary associative response (most frequent response as produced by the test persons) at relative distances between −50 and +50 words. However, to give a general picture and to abstract away from idiosyncrasies, the figure is not based on a single stimulus/response pair, but instead represents the average of 100 German stimulus/response pairs as used by Russell and Meseck [RUS 96]. The effect is in line with expectations: the closer we get to the stimulus word, the higher the chances that the primary associative response occurs. Only for distances of plus or minus one, there is an exception, but this is an artifact because content words are typically separated by function words, and among

1 http://www.jeuxdemots.org/jdm-accueil.php
2 https://www.smallworldofwords.org/en

our 100 primary responses there are no function words. In addition, test persons typically select content words only.

While such considerations are the basis underlying our work, in this chapter, the focus is on whether it is possible not only to compute the responses from the stimulus, but also to compute the stimulus from the responses. To the best of our knowledge, this has not been attempted before in a comparable (distributional semantics) framework, and therefore we are not aware of any directly related literature.

However, this task is somewhat related to the computation of associations when given several stimulus words simultaneously, which is sometimes referred to using the terms *multi-stimulus-* or *multiword associations* [RAP 08], or *remote association test* (RAT). A recent notable publication on the RAT, which gives pointers to other related works, is Smith *et al.* [SMI 13]. It applies this methodology on problems that require consideration of multiple constraints, such as choosing a job based on salary, location and work description. Another one is Griffiths *et al.* [GRI 07], which assumes that concept retrieval from memory can be facilitated by inferring the gist of a sentence, and using that gist to predict related concepts and disambiguate words. It implements this by using a topic model.

CIRCUS	FUNNY	NOSE
clown (24)	laugh (23)	face (16)
ring (10)	girl (11)	eyes (12)
elephant (6)	joke (8)	mouth (11)
tent (6)	laughter (6)	ear (10)
animals (5)	amusing (4)	eye (6)
top (5)	hilarious (4)	throat (4)
boy (4)	comic (3)	smell (3)
clowns (3)	ha ha (3)	bag (2)
horse (2)	ha-ha (3)	big (2)
horses (2)	sad (3)	handkerchief (2)

Table 4.1. *Top 10 sample associations for three stimulus words as taken from the Edinburgh Associative Thesaurus. The numbers of subjects responding with the respective word are given in brackets*

Our approach differs from this previous work in that it focuses on a related but different and particularly well-defined task. In our approach, we have eliminated all (for this particular task) unnecessary sophistication, such as *Latent Semantic Analysis* (which we used extensively in previous work) or *Topic Modeling*, resulting in a simple yet effective algorithm. For example, [GRI 07] reports 11.54% correctly predicted first associates. Rapp [RAP 08] presents a number of evaluations using various corpora and datasets, but with all results below 10%. The above-mentioned paper by Smith *et al.* [SMI 13] gives no such figures at all. In comparison, the best results presented here are at 54% (see section 4.3.3). It should be emphasized, however, that all comparisons have to be taken with caution, as there is no commonly used gold standard for this, and hence all authors used different test data, and different corpora. Note also that, in contrast to the related work, our focus is on the novel reverse association task, which gives us test data of unprecedented quality and quantity (as any word association norm can be used), but for which the previous test data is unsuitable as it relates to a somewhat different task.

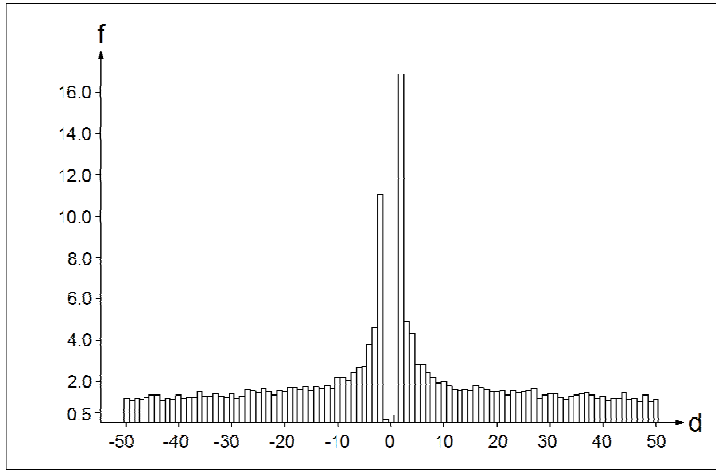


Figure 4.1. *Occurrence frequency f of a primary response at distance d from a stimulus word, averaged over 100 stimulus/response pairs [RAP 96]*

This paper, which is a substantially extended version of [RAP 13], is structured as follows: we first look at how we compute associations for

single stimulus words. This lays the basis for the second part where we reverse our viewpoint and compute the stimulus word from its associations. In the third part, we want to find out how humans perform on the reverse association task. For this purpose, we conducted a reverse association experiment where human responses were collected. Finally, we compare the performances of man and machine.

4.2. Computing forward associations

4.2.1. *Procedure*

As discussed in the introduction, we assume that there is a relationship between word associations as collected from human subjects and word co-occurrences as observed in a corpus. As our source of human data, we use the *Edinburgh Associative Thesaurus* (EAT; [KIS 73, KIS 75]), which is the largest classical collection of its kind[3]. The EAT comprises about 100 associative responses as requested from British students for each of the 8,400 stimulus terms. As some of these stimulus terms are multiword units which we did not want to include here, we removed these from the thesaurus, such that 8,210 items remained.

To obtain the required co-occurrence counts, we aimed for a corpus which is as representative as possible for the language environment of the EAT's British test subjects. We therefore chose the *British National Corpus* (BNC), a 100-million-word corpus of written and spoken language, which was compiled with the intention of providing a balanced sample of British English [BUR 98]. For our purpose, it is also an advantage that the texts in the BNC are not very recent (from 1960 to 1993), thereby including the time period when the EAT data was collected (between June 1968 and May 1971).

Since function words were not considered important for our analysis of word semantics, we decided to remove them from the text to save memory requirements and processing time. This was done on the basis of a list of approximately 200 English function words. We also decided to lemmatize

3 An even larger, though possibly more noisy, association database has been collected via online gaming at www.wordassociation.org

the corpus using the lexicon of full forms provided by Karp *et al.* [KAR 92]. This not only improves the problem of data sparseness, but also significantly reduces the size of the co-occurrence matrix to be computed. Since most word forms are unambiguous concerning their possible lemmas, we only conducted a partial lemmatization that does not take the context of a word into account and thus leaves the relatively few words with several possible lemmas unchanged. For consistency reasons, we applied the same lemmatization procedure to the whole EAT. Note that, as the EAT contains only isolated words, in this case, a lemmatization procedure that takes the context of a word into account would not be possible.

For counting word co-occurrences, as in most other studies, a fixed window size is chosen and it is determined how often each pair of words occurs within a text window of this size. Choosing a window size usually means a trade-off between two parameters: specificity versus the sparse-data problem. The smaller the window, the more salient the associative relations between the words inside the window, but the more severe the problem of data sparseness. In our case, with ±2 words, the window size looks rather small. However, this can be justified since we have reduced the effects of data sparseness by using a large corpus and by lemmatizing the corpus. It should also be noted that a window size of ±2 applied after elimination of the function words is comparable to a window size of ±4 applied to the original texts (assuming that roughly every second word is a function word).

Based on the window size of ±2, we computed the co-occurrence matrix for the corpus. By storing it as a sparse matrix, it was feasible to include all of the approximately 375,000 lemmas occurring in the BNC.

Although word associations can be successfully computed based on raw word co-occurrence counts, the results can be improved when the observed co-occurrence-frequencies are transformed by some function that reduces the effects of absolute word frequency. As it is well established, we decided to use the log-likelihood ratio [DUN 93] as our association measure. It compares the observed co-occurrence counts with the expected co-occurrence counts, thus strengthening significant word pairs and weakening incidental word pairs. In the remainder of this paper, we refer to

co-occurrence vectors and matrices that have been transformed this way as association vectors and matrices.

4.2.2. *Results and evaluation*

To compute the associations for a given stimulus word, we look at its association vector as computed in the way described above, and rank the words in the vocabulary according to association strength. Table 4.2 (two columns on the right) exemplifies the results for the stimulus word *cold*[4]. For comparison, the two columns on the left list the responses from the EAT, and words occurring in both lists are printed in bold. It can be seen that especially the test persons' most frequent responses are predicted rather well in the simulation: among the top eight experimental responses, six can be found among the computed responses.

Surprisingly, although the system solely relies on word co-occurrences, it predicts not only syntagmatic but also paradigmatic associations (e.g. not only *cold* → *ice* but also *cold* → *hot*; see [DE 96, RAP 02]).

We conducted a straightforward evaluation of the results. It is based on lemmatized versions of both the British National Corpus and, as this is the quasi-standard for evaluation in related works, the Kent and Rosanoff [KEN 10] subset of the Edinburgh Associative Thesaurus which comprises 100 words.

For 17% of the stimulus words, the system produced the primary associative response, which is the most frequent response as produced by the human subjects[5]. In comparison, the average participant in the Edinburgh Associative Thesaurus [KIS 73] produced 23.7% primary responses to these stimulus words. This means that the system performs reasonably but not quite as well as the test persons.

4 In order not to lose information, in contrast to all other results presented in this chapter, this table is based on an unlemmatized corpus and an unlemmatized association norm.

5 Wettler *et al.* [WET 05] report somewhat better results by additionally taking advantage of the observation that test persons typically answer with words from the mid-frequency range. As it is not clear how this affects the results when computing associations for several given words, we did not do so in the current paper.

Observed responses	# of subjects	Computed responses	# of subjects
hot	34	**water**	5
ice	10	**hot**	34
warm	7	weather	0
water	5	**wet**	3
freeze	3	blooded	0
wet	3	**ice**	10
feet	2	air	0
freezing	2	**winter**	2
nose	2	**freezing**	2
room	2	bitterly	0
sneeze	2	damp	0
sore	2	wind	0
winter	2	**warm**	7
arctic	1	felt	0
bad	1	**war**	1
beef	1	night	0
blanket	1	icy	0
blow	1	**heat**	1
cool	1	shivering	0
dark	1	cistern	0
drink	1	feel	0
flu	1	windy	0
flue	1	stone	0
frozen	1	morning	0
hay fever	1	shivered	0
head	1	eyes	0
heat	1	clammy	0
hell	1	sweat	0
ill	1	blood	0
north	1	shower	0
often	1	rain	0
shock	1	winds	0
shoulder	1	tap	0
snow	1	dry	0
store	1	**dark**	1
uncomfy	1	grey	0
war	1	hungry	0

Table 4.2. *Comparison between observed and computed associative responses to the stimulus word* cold *(matching words in bold; no lemmatization; capitalized words transferred to lower case)*

4.3. Computing reverse associations

4.3.1. *Problem*

Having seen that word associations with single stimulus words can be computed with a quality similar to that achieved by human subjects, let us now turn to the main question of this paper, namely whether it is also possible to reverse the task, i.e. to compute a stimulus word from its associations.

Let us look at an example: According to the Edinburgh Associative Thesaurus, the top three most frequent responses to *clown* are *circus* (produced by 26 out of 93, i.e. 28% of the test persons), *funny* (9% of the test persons) and *nose* (8% of the test persons). The question is now: given only the three words *circus*, *funny* and *nose*, is it possible to determine that their common stimulus word is *clown*? And if it is possible, what would be the quality of the results?

The above is an illustrative example, but, in other cases, it is often more difficult to guess the correct answer. To give a feeling for the difficulty of the task, let us provide a few more examples involving varying numbers of given words, with the solutions provided in Table 4.4:

apple, juice → ?

water, tub, clean → ?

grass, blue, red, yellow → ?

drink, gin, bottle, soda, Scotch → ?

4.3.2. *Procedure*

Our first idea on how to compute the stimulus given the responses was to look at the associations of the responses, and to determine their intersection. However, in preliminary experiments, we found out that this does not work well. The reason appears to be asymmetry in word association. But what do we mean by asymmetry in this context?

The co-occurrence counts that we extract from the corpus are symmetric, because whenever word A co-occurs with word B, word B also co-occurs with word A. Whether an association matrix computed from the co-occurrence matrix is also symmetric depends on the association measure used. However, even in the case of symmetric weights, associations can still be asymmetric. Let us illustrate this using Figure 4.2. This is the graphical equivalent of a symmetric association matrix[6]. As can be seen, the strongest association with *blue* is *black*. However, the opposite is not true: the strongest association with *black* is not *blue* as *black* has an even stronger association with *white*.

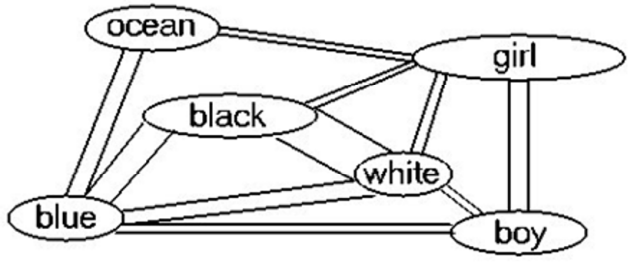


Figure 4.2. *Associative lexical network with symmetric weights*

To give an idea about the situation in the EAT: not considering multiword units, the EAT comprises 8,210 stimulus words and likewise 8,210 primary responses. However, there is not a complete overlap between these two vocabularies: only 7,387 words occur in both, which means that, only for these words, symmetry considerations are possible. Of these 7,387 cases, 63% of the responses were symmetric, and 37% were asymmetric. Table 4.3 shows some examples from the EAT for both types of associations.

6 In the asymmetric case, we would require two directed connections of (usually) different widths between each pair of nodes.

Symmetric Associations			Asymmetric Associations		
Stimulus	PR on Stimulus	PR on Response	Stimulus	PR on Stimulus	PR on Response
bed	sleep	bed	baby	boy	girl
black	white	black	bitter	sweet	sour
boy	girl	boy	comfort	chair	table
bread	butter	bread	cottage	house	home
butter	bread	butter	dream	sleep	bed
chair	table	chair	hand	foot	shoe
dark	light	dark	heavy	light	dark
girl	boy	girl	lamp	light	dark
hard	soft	hard	red	blue	sky
light	dark	light	sickness	health	wealth

Table 4.3. *Examples for symmetric and asymmetric associations (PR = primary response)*

Let us now return to our above example, namely *circus*, *nose*, *funny* → *clown*. Here, *circus* and *clown* are an example for the symmetric case. Both are each other's primary associative responses in the EAT, and therefore *circus* is the strongest association with *clown*, and likewise *clown* is the strongest association with *circus*. If this were always true, things would be straightforward. However, this is not the case. For example, *clown* is strongly associated with *nose*, but *nose* is not strongly associated with *clown*. In the EAT, among 97 test persons, given the stimulus word *nose*, none responded with *clown*. Likewise, given the word *funny*, among 98 test

persons, again nobody answered with *clown*. Therefore, if we take the intersection of the associations with *circus*, *nose*, and *funny*, *clown* would be out. This is why an approach based on intersecting associations does not work well.

Instead, like in word sense disambiguation and like in multiword semantics, it appears that we have to take contextual information into account[7]. For example, in the context of *circus*, *nose* is clearly related to *clown*, but, in the context of *doctor*, it is not.

Such considerations resulted in the following approach: we utilize the observation that a stimulus word must have strong weights to all of its top associations, and that a strong association with only some of them does not suffice. Such a behavior can usually be put into practice by using a multiplication.

However, we do not multiply the association strengths, as the log-likelihood ratio has an inappropriate (exponential) value characteristic. This value characteristic has the effect that a weak association with one of the stimuli can easily be overcompensated by a strong association with another stimulus, which is not desirable. Instead of multiplying the association strengths, we therefore multiply their ranks. This improves the results considerably.

These considerations lead us to the following procedure (see [RAP 08]): given an association matrix of vocabulary V containing the log-likelihood ratios between all possible pairs of words, to compute the stimulus word causing the responses *a*, *b, c, ...*, the following steps are conducted:

1) For each word in V (by considering its association vector), look up the ranks of words *a*, *b, c, ...* in its association vector, and compute the product of these ranks ("Product-of-Ranks algorithm").

2) Sort the words in V according to these products, with the sort order such that the lowest value obtains the top rank (i.e. conduct a reverse sort).

7 Further reflections on this may lead to the fundamental question whether asymmetry of word associations is the consequence of word ambiguity, or whether word ambiguity is the consequence of asymmetry of word associations.

Note that this procedure is somewhat time consuming as these computations are required for each word in a large vocabulary[8]. On the plus side, the procedure is in principle applicable to any number of given words, and with an increasing number of given words, there is only a slight increase in computational load.

A minor issue is the assignment of ranks to words that have identical log-likelihood scores, especially in the frequent case of zero co-occurrence counts. In such cases, the assignment of almost arbitrary ranks within such a group of words could adversely affect the results. We therefore suggest assigning corrected ranks, which are to be chosen as the average ranks of all words with identical log-likelihood scores.

In principle, the algorithm can also be used if there is only a single given word. However, this does not make much sense as the algorithm is computationally far more expensive than what we described in section 4.2, and the results are typically worse for the reason that ranks do not allow as fine-grained distinctions as do association strengths. For example, given the word *white*, the algorithm might find several words in the vocabulary where *white* is on rank 1 (e.g. *black* and *snow*). However, as (without further sophistication) no distinction is made between these, they will end up in arbitrary order, without taking into account that the top rank of *black* is more salient than that of *snow*[9].

On the other hand, if the number of given words becomes larger, depending on the application, it can be helpful to introduce a limit to the maximum rank, thereby reducing the effects of statistical variation, which is especially severe for the lower ranks. Note that for the current work, we used a rank limit of 10,000. However, the exact value is not critical because this usually has little impact if the focus is mainly on the top ranks, as is the case here.

8 Considerable time savings are possible by using an index of the non-zero co-occurrences.
9 In the EAT, 57 and 40 subjects, respectively, responded with *white* for these two stimulus words; another example is *lily*, where the primary associative response is also *white*, but is produced by only 19 subjects. In this case, the next frequent response, namely *flower*, is very close as it is produced by 17 subjects.

4.3.3. *Results and evaluation*

To give an impression of the results when applying the above algorithm to various responses from the EAT, Table 4.4 lists some results. For example, the EAT lists *apple* and *juice* as the top responses when given the stimulus word *fruit*, but our algorithm, when provided with *apple* and *juice*, computes that *orange* would be the best stimulus. This is not as expected, but also has some plausibility. The expected stimulus *fruit* at least shows up on the 8th position of the computed list of words.

For a quantitative evaluation, like for the forward associations, we consider only the Kent and Rosanoff [KEN 10] subset of the EAT[10]. We count in how many cases the expected word is ranked first in the list of computed words. This leads to conservative numbers as only exact matches are taken into account. For example, the last item in Table 4.4, where *whisky* instead of *whiskey* is on rank 1, would count as incorrect.

When predicting the stimuli from the associative responses, the question is how many of the responses should be taken into account, and in how far the quality of the results depends on the number of responses. To answer this question, we conducted the evaluation several times, each time with another number of given words (= EAT responses). There are three expectations:

– the more subjects have given a response, the more salient it is and the more helpful it should be for predicting the stimulus;

– responses given by only one or very few subjects might be arbitrary and therefore not helpful for predicting the stimulus;

– considering a larger number of salient responses should improve the results.

These expectations are confirmed by the results. Figure 4.3 shows the percentage of correctly computed stimuli depending on the number of top responses (from the EAT) that are taken into account. As can be seen, the quality of the results improves up to seven given words where it reaches 54% accuracy, and from then on degrades. This means that, on average, already the eighth response word is not helpful for determining the respective stimulus word.

10 It should be noted that the Kent and Rosanoff [KEN 10] subset typically leads to relatively high accuracies, as it mostly comprises familiar words with high corpus frequencies.

TOP 2 RESPONSES FROM EAT: apple (1,385) juice (1,613)
STIMULUS WORD FROM EAT: **fruit** (3,978)
COMPUTED STIMULI: orange (2,333), grape (273), lemon (1,019), lime (612), pineapple (220), grated (423), apples (792), **fruit** (3,978), grapefruit (113), carrot (359)

TOP 3 RESPONSES FROM EAT: water (33,449), tub (332), clean (6,599)
STIMULUS WORD FROM EAT: **bath** (415)
COMPUTED STIMULI: rinsed (177), **bath** (2,819), soak (315), rinse (288), wash (2,449), refill (138), rainwater (160), polluted (393), towels (421), sanitation (156)

TOP 4 RESPONSES FROM EAT: grass (4,295), blue (9,986), red (13,528), yellow (4,432)
STIMULUS WORD FROM EAT: **green** (10,606)
COMPUTED STIMULI: **green** (10,606), jersey (359), ochre (124), bright (5,313), pale (3,583), violet (396), purple (1,262), greenish (136), stripe (191), veined (103)

TOP 5 RESPONSES FROM EAT: drink (7,894), gin (507), bottle (4,299), soda (356), Scotch (621)
STIMULUS WORD FROM EAT: **whiskey** (129)
COMPUTED STIMULI: whisky (1,451), **whiskey** (129), tonic (511), vodka (303), brandy (848), Whisky (276), scotch (151), lemonade (229), poured (1,793), gulp (196)

Table 4.4. *Top 10 computed stimuli for various numbers of given responses. Numbers in brackets refer to corpus frequencies in the BNC*

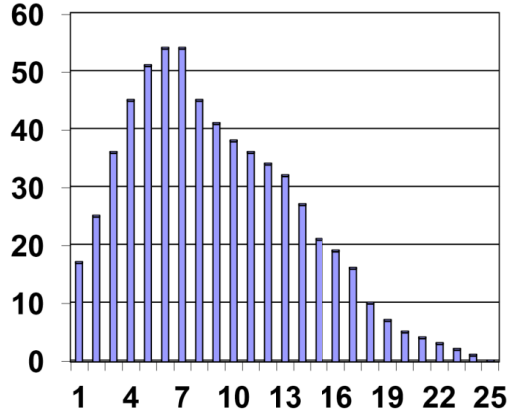


Figure 4.3. *Percentage of correctly predicted stimuli (vertical axis) depending on the number of given words (horizontal axis)*

Let us mention that we were positively surprised by the 54% performance figure, which is about three times as good as for forward associations (first column in Figure 4.3). On the one hand, in the reverse association task, there are several clues pointing to the same stimulus word. On the other hand, the task seems non-trivial for humans, and typically there are several plausible options how the given words can disambiguate each other. For example, given *apple* and *juice* (see Table 4.4), the solution our system came up with, namely *orange*, seems quite as plausible as the expected solution *fruit*. However, in our evaluation, *orange* is counted as wrong, and this is true for many others of the 46% incorrect results.

4.4. Human performance

Now that we have investigated machine performance on the reverse association task, the next question is how humans perform on this task, and how the results of a human and a machine compare. For this purpose, we conducted an experiment with the aim of collecting human reverse associations, and later on compared its results with those obtained in a simulation.

4.4.1. *Dataset*

As our dataset we used the test set which – as a follow-up activity of [RAP 13] – we had prepared for the *CogALex-IV Shared Task on the Lexical Access Problem* [RAP 14]. The aim of this shared task had been to compare different automatic methods for computing reverse associations[11]. In contrast, here we investigate human performance on this task. Therefore, to compare human with machine associations, it is good that both studies use a dataset based on the same source, i.e. on the EAT[12].

11 The best performing systems in this shared tasks all used methodologies similar to the system described in this paper (i.e. based on word co-occurrences in large corpora) and their results were of similar quality. As they have been discussed in detail in [RAP 13], we do not repeat this discussion here.

12 Note that, in addition to the dataset described here, the shared task at CogALex-IV used an additional so-called "training set", which was intended for the development and optimization of the automatic algorithms. This training set was produced in exactly the same way, just using a different selection of 2,000 items.

The dataset has been produced by conducting the following steps:

1) take the EAT as the basis;

2) modify capitalization;

3) extract 2,000-item subset (mainly at random);

4) retain only the top five associations for each stimulus word;

5) remove stimulus words.

Let us now look a bit closer at this procedure. The EAT lists for each of the 8,400 stimulus words the associative responses as obtained from about 100 test persons[13] who were asked to produce the word coming spontaneously to their mind.

As, given its origin in the 1970s, the EAT uses uppercase characters only, we decided to modify its capitalization. For this purpose, for each word occurring in the EAT, we looked up which form of capitalization showed the highest occurrence frequency in the *British National Corpus* [BUR 98]. By this form, we replaced the respective word, e.g. *DOOR* was replaced by *door*, and *GOD* was replaced by *God*. This way we hoped to come close to what might have been produced during compilation of the EAT if case distinctions had been taken into account. Since this method is not perfect, e.g. words often occurring in the initial position of a sentence might be falsely capitalized, we did some manual checking, but cannot claim to have achieved perfection.

For each stimulus word, only the top five associations (i.e. the associations produced by the largest numbers of test persons) were retained, and all other associations were discarded. The decision to keep only a small number of associations was motivated by the results shown in Figure 4.3, which indicate that associations produced by only very few test persons tend to be of an arbitrary nature. We also wanted to avoid unnecessary complications, which is why we decided on a fixed number, although the choice of exactly five associations is somewhat arbitrary.

13 To distribute work, different groups of test persons operated on the stimulus words.

Given words	Target words
able incapable brown clever good	capable
able knowledge skill clever can	ability
about near nearly almost roughly	approximately
above earth clouds God skies	heavens
above meditation crosses passes rises	transcends
abuse wrong bad destroy use	misuse
accusative calling case Latin nominative	vocative
ache courage blood stomach intestine	guts
ache nail dentist pick paste	tooth
aches hurt agony stomach period	pains
action arc knee reaction jerk	reflex
actor theatre door coach act	stage

Table 4.5. *Top 12 items of the dataset. The target words were of course undisclosed to the test takers*

From the remaining dataset, we removed all items that contained non-alphabetical characters. We also removed items that contained words that did not occur in the BNC. The reason for this is that quite a few of them are misspellings. By these measures, the number of EAT items was reduced from initially 8,400 to 7,416. From these, we randomly selected 2,000 items that were used as our dataset. Table 4.5 shows the (alphabetically) first 12 items in this dataset[14].

4.4.2. *Test procedure*

As, according to comments from our test persons, producing multi-stimulus associations is subjectively hard, in order not to overload the test takers, we divided the dataset into 40 sections, each comprising 50 items. A total of 40 test sheets were printed, each showing the test items from one section. The test takers were instructed to produce for each test item of five words the spontaneous association that first came to mind. It was also

14 The full dataset can be downloaded from http://pageperso.lif.univ-mrs.fr/~michael.zock/ColingWorkshops/CogALex-4/co-galex-webpage/pst.html

mentioned that there are no "right" or "wrong" answers, but that we were only surveying human word associations[15].

As an example, the list "*work desk bureau secretary post*" was provided, together with the response "*office*". As the experiment was conducted at the University of Mainz, Faculty of Translation Studies, Linguistics and Cultural Studies (FTSK) in Germersheim (Germany), all participants were non-native English speakers. For this reason, they were asked to rank their English proficiency on the following scale: *native / very good / good / satisfactory / basic knowledge / no or very little knowledge*. Note, however, that because the FTSK is specialized in training translators and interpreters, of whom many focus on English, as expected the majority of students indicated "very good" English proficiency, and none indicated a proficiency below "satisfactory". The experiments were conducted in the winter term of 2014/15 and in the summer term of 2015 in the courses given by the author. Course topics were *Translation Systems, Language and Cognition, Computational Linguistics, Electronic Dictionaries, Corpus Linguistics, Machine Translation, Computer Aided Translation* and *Translation Memories* (mostly seminars at either undergraduate or graduate level)[16]. Participation in the association test was voluntary. As the task was perceived to be tedious, many questionnaires showed some omissions. Altogether, 66 questionnaires were filled out, which implies that some of the 40 sections of the dataset were dealt with by more than one student.

4.4.3. *Evaluation*

For evaluation, we simply compared the associations produced by the test persons with the expected results as taken from the EAT (see Table 4.2). In a first round of evaluation, we only considered exact matches as correct, with the only flexibility that word capitalization was not taken into account. However, in a second round, we also counted inflectional variants and derived forms of the expected word as correct. For example, for the given words "*car round cart spoke bicycle*", a test person responded with *wheels*, whereas the expected solution was *wheel*. This was now also considered to be a match.

15 Note that we did not inform the participants about the nature of the test data, i.e. that the underlying idea is based on the reverse association task.
16 For the purpose of this paper, German course titles were translated into English.

English proficiency	exact matches	tolerant matches
very good (48 subjects)	4.38 %	7.42 %
good (14 subjects)	2.14 %	4.29 %
satisfactory (4 subjects)	4.50 %	7.50 %
All 66 subjects	3.91 %	6.76 %

Table 4.6. *Results by language proficiency for the two modes of evaluation*

The overall results are shown in Table 4.6. For each proficiency group, the percentage of correctly predicted words is specified for the two modes of evaluation. All accuracies are in a range between 2 and 8%. As can be seen, the students who rated their language proficiency to be very good did considerably better than the ones who rated themselves to be only good; but surprisingly, the students with an only satisfactory self-rating did even better. Note, however, that this is a small group of only four subjects who cannot be taken as representative (and whose self-ratings might simply reflect modesty).

4.5. Performance by machine

Although we have presented simulation results concerning the reverse association task in section 4.3, these results were obtained using a much smaller test set than the one used for the human survey. We therefore conducted an additional system run using the human test data. The parameters were the same as described previously (section 4.2). However, as our corpus we did not use the BNC but rather ukWaC, a web-derived corpus of about 2 billion words[17]. This has the advantage that our results can also be compared with those of the CogALex shared task [RAP 14], where the ukWaC corpus has also been used.

As done previously for the BNC, we lemmatized the ukWaC corpus and removed stop words. As our vocabulary we used a list of words which in the BNC had an occurrence frequency of 100 or higher. According to our definition of a word (string of alpha or of non-alpha characters), this was the case for 36,097 words. However, like the ukWaC corpus, we also

17 http://wacky.sslmit.unibo.it/doku.php?id=corpora

lemmatized this list and removed stop words. We also eliminated any strings containing non-alpha characters. This reduced the vocabulary to 22,578 words.

Using this list, by counting word co-occurrences in the ukWaC-corpus, we built up a co-occurrence matrix, thereby considering a window size of plus and minus two words around a given word. The resulting co-occurrence matrix was converted into a weight matrix by applying the log-likelihood ratio [DUN 93] to each value in the matrix. Finally, the product-of-ranks algorithm was applied to compute the results for each item of five words in the test set.

Of the 2,000 items, the system got 613 right, i.e. the word on rank 1 of the computed list was exactly identical to the expected word as provided in the test set. This corresponds to an accuracy of 30.65%. Despite the larger corpus (ukWaC is about 20 times larger than the BNC), this accuracy is considerably lower than the top performance figure of 54% reported in section 4.3. However, the discrepancy is not surprising as, in section 4.3, the test was conducted with the 100 words from the Kent–Rosanoff test. This test contains mostly very common words well known for their salient associations, which are comparatively easy to predict. Also, in section 4.3, the words in the test set had been lemmatized, which is not the case here.

Therefore, let us compare our results with those of the CogALex shared task [RAP 14]. There, on exactly the same test set and based on the same corpus, the best system showed a performance of 30.45%, which almost exactly matches the performance of our system. However, while this system used sophisticated technology involving word embeddings (neural network technology), we achieved a similar result using very simple technology. It should also be noted that we did not do any parameter optimization and simply used the parameters from a previous paper [RAP 13], i.e. to obtain the current results, we did not even look at the training set, which was also provided for the CogALex shared task.

It should also be noted that our selection of vocabulary was completely unrelated to the current task. Our word list was derived from the BNC rather than from ukWaC, and the fact that the test set was derived from the EAT had no influence on our choice of words[18]. We also used an arbitrary

18 Would the vocabulary be derived from the EAT, better results could be expected.

frequency threshold (BNC frequency of 100) for the selection of our vocabulary, rather than trying to optimize this threshold using the training set.

It should be noted that the choice of vocabulary is very important for this task, and much better results could be achieved by applying "informed guesses" on this issue. Let us mention that of the 2,000 unique solutions from the test set[19], in our vocabulary of 22,755 words, only 1,482 occurred, i.e. for 518 words, our system had no chance to come up with the correct solution.

4.6. Discussion, conclusions and outlook

4.6.1. *Reverse associations by a human*

Associating with several given words, as required in the reverse association task, is not easy. This has been remarked by several test persons and is also confirmed by a considerable number of omissions in the test sheets. For several reasons, it is difficult to come up with an expected word:

1) In many cases, the given words might almost quite as strongly point to other target words. For example, when given the words *gin*, *drink*, *scotch*, *bottle* and *soda*, instead of the target word *whisky*, the alternative spelling *whiskey* should also be fine, and possibly some other alcoholic beverages, such as *rum* or *vodka*, might also be acceptable[20].

2) The target vocabulary was not restricted in any way, so in principle hundred thousands of words had to be considered as candidates.

3) Although most of the target words were base forms, the dataset also contains a good number of cases where the target words were inflected forms. Of course, it is hard to get these inflected forms exactly right.

Owing to these difficulties, we expected low-performance figures, and this expectation was confirmed.

19 Each solution can occur just once because they are derived from the EAT stimuli.

20 As our data source (the EAT) did not provide any, it was not practical for us to try to come up with alternative solutions in the chosen reverse association framework. However, we think that doing so would mainly affect absolute but not relative performance (i.e. the ranking between different systems should remain similar).

As mentioned in section 4.4.2, we had not disclosed the nature of the dataset to the test subjects, i.e. did not tell them that the underlying idea is the reverse association task. Alternatively, we could have informed people about this and asked them to come up with a word with which they would associate each of the five given words. But this would probably have been conceived as an even more sophisticated task, potentially blocking spontaneity. We also tend to think that, although associations can occasionally be asymmetric[21], assymetry is likely not to have a decisive effect in our scenario. However, this is certainly a question which requires further investigation

While in previous studies involving multi-stimulus associations, human performance had always been much better than what simulation programs produced (see e.g. [RAP 08]), this is not the case here. In the CogALex shared task, a number of teams had tested their algorithms, which were mostly based on the analysis of word co-occurrences in large text corpora, on the very same dataset, and achieved performances of up to 30.45%. In the current paper, on this dataset, we presented a very similar result. These results are much better than the human performance of our non-native speakers shown in Table 4.6. Although native speakers can be expected to do better, we think that it will be challenging to outperform the automatic results. This might well mean that in one of its core disciplines, namely association, human intelligence is not better than a machine, although of course further investigation is required to confirm this finding.

4.6.2. *Reverse associations by a machine*

We introduced the product-of-ranks algorithm and showed that it can be successfully applied to the problem of computing associations if several words are given. To evaluate the algorithm, we used the EAT as our gold standard, but assumed that it makes sense to look at this data in the reverse direction, i.e. to predict the EAT stimuli from the EAT responses.

Although this is a task even difficult for humans, and although we applied a conservative evaluation measure that insists on exact string matches between a predicted and a gold standard association, our algorithm was able to do so with a success rate of approximately 30% (54% for the

21 For example, *flower pot* → *soil* are associated strongly, but *soil* → *flower pot* not to the same extent.

Kent–Rosanoff vocabulary). We also showed that, up to a certain limit, with increasing numbers of given words, the performance of the algorithm improves, and only thereafter degrades. The degradation is in line with our expectations because associative responses produced by only one or very few persons are often of almost arbitrary nature and therefore not helpful for predicting the stimulus word[22].

Given the notorious difficulty to predict experimental human data, we think that the performance of approximately 30% is quite good, especially in comparison to the human results shown in Table 4.6, but also in comparison to the related work mentioned in the introduction (11.54%), and to the results on single stimuli (17%). However, there is of course still room for improvement, even without moving to more sophisticated (but also more controversial) evaluation methods that allow alternative solutions. We intend to advance from the product-of-rank algorithm to a product-of-weights algorithm. But, this requires that we have a high-quality association measure with an appropriate value characteristic. One idea is to replace the log-likelihood scores by their significance levels. Another is to abandon conventional association measures and move on to empirical association measures as described in Tamir and Rapp [TAM 03]. These do not make any presuppositions on the distribution of words, but determine this distribution from the corpus. In any case, the current framework is well suited for measuring and comparing the suitability of any association measure. Further improvements might be possible by using neural vector space models (word embeddings), as investigated by some of the participants of the CogALex-IV shared task [RAP 14].

Concerning applications, we see a number of possibilities: one is the tip-of-the-tongue problem, where a person cannot recall a particular word but can nevertheless think of some of its properties and associations. In this case, descriptors for the properties and associations could be fed into the system in the hope that the target word comes up as one of the top associations, from which the person can choose.

Another application is in information retrieval, where the system can help to sensibly expand a given list of search words, which is in turn used to conduct a search. A more ambitious (but computationally expensive) approach would be to consider the (salient words in the) documents to be

22 Such associations might reflect very specific experiences of a test person.

retrieved as our lists of given words, and to predict the search words from these using the product-of-ranks algorithm.

A further application is in multiword semantics. Here, a fundamental question is whether a particular multiword expression is of compositional or of contextual nature. The current system could possibly help to provide a number of quantitative measures relevant for answering the following questions:

1) Can the components of a multiword unit predict each other?

2) Can each component of a multiword unit be predicted from its surrounding content words?

3) Can the full multiword unit be predicted from its surrounding content words?

The results of these questions might help us to answer the question regarding a multiword unit's compositional or contextual nature, and to classify various types of multiword units.

The last application we would like to propose here is natural language generation (or any application that requires it, e.g. machine translation or speech recognition). If in a sentence, one word is missing or uncertain, we can try to predict this word by considering all other content words in the sentence (or a somewhat wider context) as our input to the product-of-ranks algorithm.

From a cognitive perspective, the hope is that such experiments might lead to some progress in finding an answer concerning a fundamental question: is human language generation governed by associations, i.e. can the next content word of an utterance be considered as an association with the representations of the content words already activated in the speaker's memory?

4.7. Acknowledgments

This research was supported by a Marie Curie Career Integration Grant within the 7th European Community Framework Programme. Many thanks to the students who participated in the association experiments.

4.8. Bibliography

[BUR 98] BURNARD L., ASTON G., *The BNC Handbook: Exploring the British National Corpus*, Edinburgh University Press, Edinburgh, 1998.

[CHU 90] CHURCH K.W., HANKS P., "Word association norms, mutual information, and lexicography", *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[DE 96] DE SAUSSURE F., *Cours de linguistique générale*, Payot, Paris, 1996.

[DUN 93] DUNNING T., "Accurate methods for the statistics of surprise and coincidence", *Computational Linguistics*, vol. 19, no. 1, pp. 61–74, 1993.

[GRI 07] GRIFFITHS T.L., STEYVERS M., TENENBAUM J.B., "Topics in semantic representation", *Psychological Review*, vol. 114, no. 2, pp. 211–244, 2007.

[JAM 90] JAMES W., *The Principles of Psychology*, Holt, New York, 1890.

[JEN 70] JENKINS J., "The 1952 Minnesota word association norms", in POSTMAN L., KEPPEL G. (eds), *Norms of Word Association*, Academic Press, New York, 1970.

[KAR 92] KARP D., SCHABES Y., ZAIDEL M. *et al.*, "A freely available wide coverage morphological analyzer for English", *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, pp. 950–955, 1992.

[KEN 10] KENT G.H., ROSANOFF A.J., "A study of association in insanity", *American Journal of Psychiatry*, vol. 67, pp. 317–390, 1910.

[KIS 73] KISS G.R., ARMSTRONG C., MILROY R. *et al.*, "An associative thesaurus of English and its computer analysis", in AITKEN A., BAILEY R., HAMILTON-SMITH N. (eds), *The Computer and Literary Studies*, Edinburgh University Press, 1973.

[KIS 75] KISS G.R., "An associative thesaurus of English: structural analysis of a large relevance network", in KENNEDY A., WILKES A. (eds), *Studies in Long Term Memory*, Wiley, London, 1975.

[NEL 98] NELSON D., MCEVOY C., SCHREIBER T.A., "The University of South Florida word association, rhyme, and word fragment norms", available at: http://www.usf.edu/FreeAssociation, 1998.

[PAL 64] PALERMO D.S., JENKINS J.J., *Word Association Norms: Grade School Through College*, University of Minnesota Press, Minneapolis, 1964.

[RAP 96] RAPP R., *Die Berechnung von Assoziationen*, Olms, Hildesheim, 1996.

[RAP 02] RAPP R., "The computation of word associations: comparing syntagmatic and paradigmatic approaches", *Proceedings of the 19th International Conference on Computational Linguistics*, Taipeh, vol. 2, pp. 821–827, 2002.

[RAP 08] RAPP R., "The computation of associative responses to multiword stimuli", *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, Manchester, pp. 102–109, 2008.

[RAP 13] RAPP R., "From stimulus to associations and back", *Proceedings of the 10th Workshop on Natural Language Processing and Cognitive Science*, pp. 78–91, Marseille, France, 2013.

[RAP 14] RAPP R., ZOCK M., "The CogALex-IV shared task on the lexical access problem", *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon*, Dublin, Ireland, pp. 1–14, 2014.

[RUS 96] RUSSELL W.A., MESECK O.R., "Der Einfluß der Assoziation auf das Erinnern von Worten in der deutschen, französischen und englischen Sprache", *Zeitschrift für experimentelle und angewandte Psychologie*, vol. 6, pp. 191–211, 1959.

[SCH 89] SCHVANEVELDT R.W., DURSO F.T., DEARHOLT D.W., "Network structures in proxymity data", in BOWER G. (ed.), *The Psychology of Learning and Motivation: Advances in Research and Theory*, vol. 24, Academic Press, New York, 1989.

[SMI 13] SMITH K.A., HUBER D.E., VUL E., "Multiply-constrained semantic search in the Remote Associates Test", *Cognition*, vol. 128, pp. 64–75, 2013.

[TAM 03] TAMIR R., RAPP R., "Mining the web to discover the meanings of an ambiguous word", *Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, FL, pp. 645–648, 2003.

[WET 89] WETTLER M., RAPP R., "A connectionist system to simulate lexical decisions in information retrieval", in PFEIFER R., SCHRETER Z., FOGELMAN F. *et al.* (eds), *Connectionism in Perspective*, Elsevier, Amsterdam, 1989.

[WET 05] WETTLER M., RAPP R., SEDLMEIER P., "Free word associations correspond to contiguities between words in texts", *Journal of Quantitative Linguistics*, vol. 12, no. 2, pp. 111–122, 2005.

This page intentionally left blank

5

Hidden Structure and Function in the Lexicon

How many words (and which ones) are needed to define all the rest of the words in a dictionary? We applied graph theoretic analysis to the Wordsmyth suite of dictionaries. By recursively removing every word that is defined but defines no further words, every dictionary can be reduced to a small subset of words, the *Kernel*, which can define all the rest of the words in the dictionary, including one another; but the Kernel, though unique, is not the smallest subset that can define all the rest. The Kernel consists of one huge strongly connected component (SCC), the *Core*, about three quarters the size of the Kernel, surrounded by many tiny SCCs, the Satellites. Core words can define one another but cannot define the rest of the dictionary. The smallest number of words that can define all the rest of the dictionary is the "minimum feedback vertex set" or *Minimal Grounding Set (MinSet)*. MinSets are not unique. Each dictionary's Kernel contains many overlapping MinSets. About a fifth of the size of the Kernel, each MinSet is part-Core, part-Satellites; its words can define all the rest of the dictionary, but not one another. The Core words are more frequent, learned earlier and less concrete than the Satellite words, which are in turn more frequent and learned earlier but more concrete than the rest of the dictionary. We will discuss implications for language learning, language evolution and the representation of word meaning in the mental lexicon.

5.1. Introduction

Dictionaries catalogue and define the words of a language[1]. In principle, since every word in a dictionary is defined, it should be possible to learn the

Chapter written by Philippe VINCENT-LAMARRE, Mélanie LORD, Alexandre BLONDIN-MASSÉ, Odile MARCOTTE, Marcos LOPES, Stevan HARNAD.

1 Almost all the words in a dictionary (whether nouns, verbs, adjectives or adverbs) are "content" words, i.e. they are the names of categories [HAR 05]. Categories are kinds of things, both concrete and abstract (objects, properties, actions, events, states). The only words that are not the names of categories are logical and grammatical "function" words such as if, is, the, and, not. Our analysis is based solely on the content words; function ("stop") words are omitted.

meaning of any word through verbal definitions alone [BLO 13]. However, in order to understand the meaning of the word that is being defined, we have to understand the meaning of the words used to define it. If not, we have to look up the definition of those words too. However, if we have to keep looking up the definition of each of the words used to define a word, and then the definition of each of the words that define the words that define the words, and so on, we will eventually come full circle, never having learned a meaning at all.

This is the symbol grounding problem: the meanings of all words cannot be learned through definitions alone [HAR 90]. The meanings of some words, at least, have to be "grounded" by some means other than verbal definitions. That other means is direct sensorimotor experience with the referent of the word [HAR 10, PÉR 16], but the learning of categories from sensorimotor experience is not the subject of this paper. Here, we ask only how many words need to be grounded by some means other than verbal definition such that all the rest can be learned via definitions composed out of only those already grounded words -- and how do those grounding words differ from the rest?

5.2. Methods

5.2.1. *Dictionary graphs*

To answer this question, dictionaries can be analyzed using graph theory. We have previously analyzed several English dictionaries (including Longman's, Cambridge, Merriam-Webster and WordNet; [VIN 16]). In the present paper, we replicate and extend these results for the Wordsmyth Advanced Dictionary-Thesaurus (WADT, 70K words) as well as for three reduced versions of it [Wordsmyth Children's Dictionary-Thesaurus (WCDT, 20K words), Wordsmyth Beginner's Dictionary-Thesaurus (WBDT, 6K words), and Wordsmyth Illustrated Learner's Dictionary (WILD, 4K words)], modified for specific purposes by using fewer and more frequent words [PAR 98]: http://www.wordsmyth.net.

Apart from proper names, all words used in a dictionary are defined. In a dictionary graph, there is a directional link from each defining word to each defined word. Our analyses of dictionary graphs have revealed a hidden structure in dictionaries, which has not been observed or reported previously (see Figure 5.1). If we remove recursively all those words that are defined but

that do not go on to define any further word, every dictionary graph can be reduced by about 85% (the percentage reducibility becomes less, the smaller the dictionary) to a unique set of words (which we have called the Kernel, K), out of which all the words in the dictionary can be defined [BLO 08]. There is only one such Kernel in any dictionary, but the Kernel is not the *smallest* number of words, *M*, out of which all the words in the dictionary can be defined. That smallest number of words is the dictionary graph's "minimum feedback vertex set" (see, e.g., [FOM 08, KAR 72, LAP 12], which we will call the dictionary's *MinSet (Minimal Grounding Set)*.

For the Wordsmyth dictionaries analyzed in this paper, the relative size of the Kernel varies from 17% of the whole dictionary in the largest dictionary (WADT) to 49% in the smallest (WILD). In all four dictionaries, the MinSet size is about a fifth of the Kernel (Table 5.1). Unlike the Kernel, however, the MinSet is not unique: There are a huge number of (overlapping) MinSets in every dictionary, each of the same minimal size, M. Each is a subset of the Kernel and any one of them, if it were grounded, would then ground the entire dictionary[2].

	WADT	WCDT	WBDT	WILD
Total word meanings	73,158	20,129	6,038	4,244
First word meanings	43,363	11,626	4,456	3,159
Rest	36,196 (83%)	8,617 (74%)	3,254 (73%)	1,608 (51%)
Kernel	7,167 (17%)	3,009 (26%)	1,202 (27%)	1,551 (49%)
Satellites	1,999 (5%)	609 (5%)	258 (6%)	159 (5%)
Core	5,168 (12%)	2,400 (21%)	944 (21%)	1,392 (44%)
MinSets	1,335 (3%)	548 (4.7%)	234 (5.3%)	330 (10.4%)
Satellite-MinSets	569 (1.3%)	160 (1.4%)	67 (1.5%)	39 (1.2%)
Core-MinSets	766 (1.7%)	388 (3.3%)	167 (3.8%)	291 (9.2%)

Table 5.1. *The Wordsmyth dictionaries. The Kernel's Core (its biggest SCC) is always much larger than the Satellites (small SCCs), and each MinSet (part-Core, part-Satellites) is about a fifth of the size of the Kernel. [Wordsmyth Advanced Dictionary-Thesaurus (WADT, 70K words), Wordsmyth Children's Dictionary-Thesaurus (WCDT, 20K words), Wordsmyth Beginner's Dictionary-Thesaurus (WBDT, 6K words) and Wordsmyth Illustrated Learner's Dictionary (WILD, 4K words)]; Parks et al. [PAR 98])*

2 There may be something informative in an analogy between all the MinSets as potential bases for all of semantic space and the infinite number of different sets of M linearly independent vectors that can serve as the potential bases for M-dimensional vector space.

The Kernel, however, is not just a large number of overlapping MinSets. It has structure too. It consists of a large number of strongly connected components (SCCs). (A directed graph -- in which a directional link indicates that word W_1 is part of the definition of word W_2 -- is "strongly connected" if any word in the graph can be reached by a chain of definitional links from any other word in the graph.) Most of the SCCs of the Dictionary's Kernel are small, but in every dictionary, we have analyzed so far that there also turns out to be one very large SCC, about half the size of the Kernel. We call this the Kernel's Core (C)[3].

The Kernel itself is a self-contained dictionary, just as is the dictionary as a whole (D). K is a sub-dictionary of D. Every word in the Kernel can be fully defined using only words in the Kernel. The Core is likewise a self-contained dictionary; but, in all the full-size dictionaries of natural languages that we have so far examined,[4] the Kernel is not an SCC: Within the Core (but not the Kernel), every word can be reached by a chain of definitions from any other word in the Core.

The Kernel is a Grounding Set for the dictionary as a whole, but it is not a *Minimal* Grounding Set (MinSet), i.e. not the smallest number of words capable of defining all the rest of the dictionary. The Kernel's Core is not only a MinSet for the dictionary as a whole: it is not even a Grounding Set at all. The words in the Core alone are not enough to define all the rest of the words in the dictionary, outside the Core.

In contrast, all the MinSets are, like the Core, contained entirely within the Kernel, but no MinSet is completely contained within the Core: each straddles the Core and the Satellites. Each MinSet can define all the rest of the words in the dictionary outside the MinSet, but no MinSet is an SCC: its words are not even connected (Table 5.2). Indeed, the MinSet cannot define any of the words *within* the MinSet: only the words *outside* the MinSet.

---

3 Formally, the Core is defined as the union of all the strongly connected components (SCCs) of the Kernel that do not receive any incoming definitional links from outside themselves. (In graph theoretic language: there is no incoming link into the Core, i.e. no definitional link from a word not in the Core to a word in the Core.) It turns out to be an empirical fact about all the full-sized dictionaries we have analyzed so far, however, that their Core is itself always an SCC, and also by far the largest of the SCCs in the Kernel, the rest of which look like many small satellites surrounding one big planet (Figures 5.1 and 5.2).

4 In some of the mini dictionaries generated in our online dictionary game, however, the Core is not an SCC, but a disjoint union of SCCs (Figures 5.5 and 5.6).

The MinSets of Dictionaries hence turn out empirically[5] to consist of two parts: words in the Core (C) (which is entirely within the Kernel) and words in the remainder of the Kernel (K); the Satellites (S). The MinSet S/C ratio varies across dictionaries, but, within a given dictionary, it is the same for all of its MinSets. The MinSets, the smallest subsets capable of defining all the rest of the dictionary, are hence part-Core and part-Satellite. The natural question, then, is: *What, if anything, is the difference between the kinds of words that are in the various components of this hidden structure of the dictionary: the MinSets, the Core, the Satellites, the Kernel, and the rest of the dictionary outside the Kernel?*

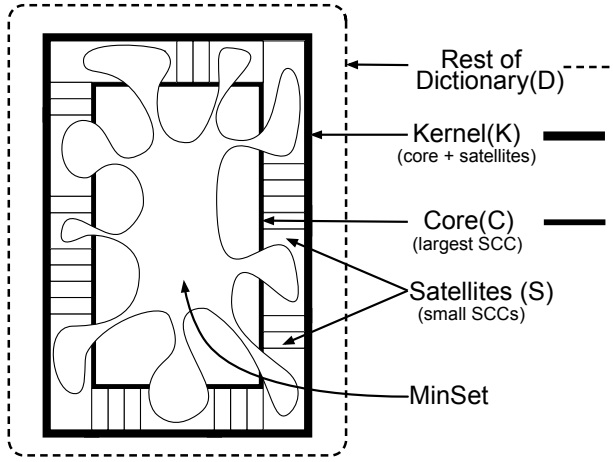


Figure 5.1. *Schematic diagram of the hidden structure of the Wordsmyth dictionaries: the Rest of the Dictionary, Kernel, Core and MinSets (only one shown). The words in these different structural components of the dictionary graph tend to differ in their psycholinguistic properties (Figures 5.2 and 5.3). Satellite words are more frequent than the words in the Rest of the Dictionary, more concrete and learned at a significantly younger age. The Core words, in turn, are even younger and more frequent than the Satellite words, but they are the Satellite words that are most concrete ones than the Core and the Rest of the Dictionary (Figure 5.2). (Note that the diagram is not drawn to scale: the relative size of the Kernel varies from 17% of the whole dictionary for the largest Wordsmyth dictionary (WADT) to 49% for the smallest (WILD))*

5 Most of the properties described here are empirically observed properties of dictionary graphs, and not necessary properties of directed graphs in general.

Is a column entry (right) necessarily a row entry (below)?	Dict	Kern	GS	SCC	Core	MinSet
Dictionary (D)	yes	yes	–	–	yes	–
Grounding Set (GS)	yes	yes	yes	–	–	yes
Strongly Connected Component (SCC)	–	–	–	yes	yes	–
Minimal Grounding Set (MinSet)	–	–	–	–	–	yes

Table 5.2. *Table indicating which of the hidden structures are Dictionaries, Grounding Sets, Strongly Connected Components and Minimal Grounding Sets*

5.2.2. *Psycholinguistic variables*

We used three databases of psycholinguistic correlates of words: age of acquisition, concreteness and word frequency. For age of acquisition, we used Kuperman *et al.*'s [KUP 12] age-of-acquisition ratings for 30,000 English words. For concreteness, we used Brysbaert *et al's.* [BRY 14] database for 40,000 common English word lemmas. For frequency, we used the $SUBTLEX_{US}$ Corpus [BRY 09]. The vast majority of these words had frequencies of less than 1,000, but a small percentage of word frequencies ranged from 5,000 to 2 million, heavily skewing the distribution. To avoid a disproportionate influence from these extreme values, we used the log base 10 of the raw frequency + 1. Due to the large size of these databases, we were able to achieve a coverage of over 90% for age and concreteness. Words missing from the corpus used in $SUBTLEX_{US}$ had frequency zero, which implies that we had frequency coverage for all words.

5.2.3. *Data analysis*

We did not apply statistical tests to the comparison between structures because we have access to all the words in each dictionary. Hence, our datasets are not samples from each dictionary: they are the entire population, making statistical estimation supererogatory. In addition, the number of observations for each hidden structure is so large that almost any marginal difference would yield statistically significant results. Thus, we rely on the replication of our observed pattern of results across all the individual dictionaries we have studied as confirmation of the generality of the findings.

5.3. Psycholinguistic properties of Kernel, Satellites, Core, MinSets and the rest of each dictionary

The words in the Kernel (K) differ significantly from words in the rest of the Dictionary (D) for all three psycholinguistic variables: Kernel words are learned significantly younger, more concrete and more frequent than the words in the rest of the dictionary. The same effect was found in comparing Core (C) words with D words for age and frequency, but not for concreteness, where only the Satellite words (S) are more concrete than C and D. Hence, the effects get stronger as we move inward from the rest of the Dictionary to the Kernel to the Core for age and frequency (but not concreteness), as schematized in the left part of Figure 5.2. There were only two small exceptions to the overall pattern in the two smallest dictionaries: for WCDT, S was not more concrete than C and D as in the other three dictionaries, and for WILD, D was not less frequent than S. Apart from these two exceptions, our results replicated the patterns observed previously for four other dictionaries [VIN 16].

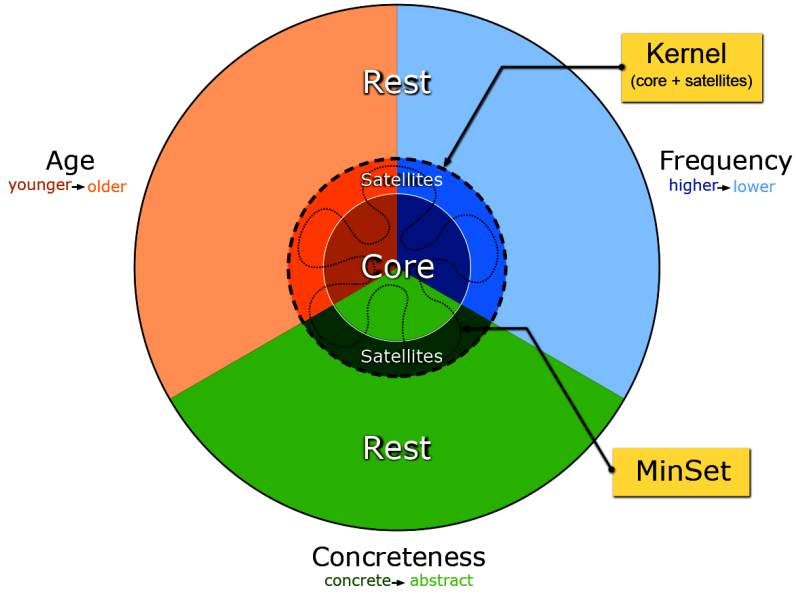


Figure 5.2. *Moving outward from the Core (C) to the Satellite (S) layer of the Kernel (K) to the rest of the Dictionary (D, about 80%), words are increasingly frequent and younger (as indicated by darkness) (see Figure 5.1 for arrows identifying each of these structures). For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

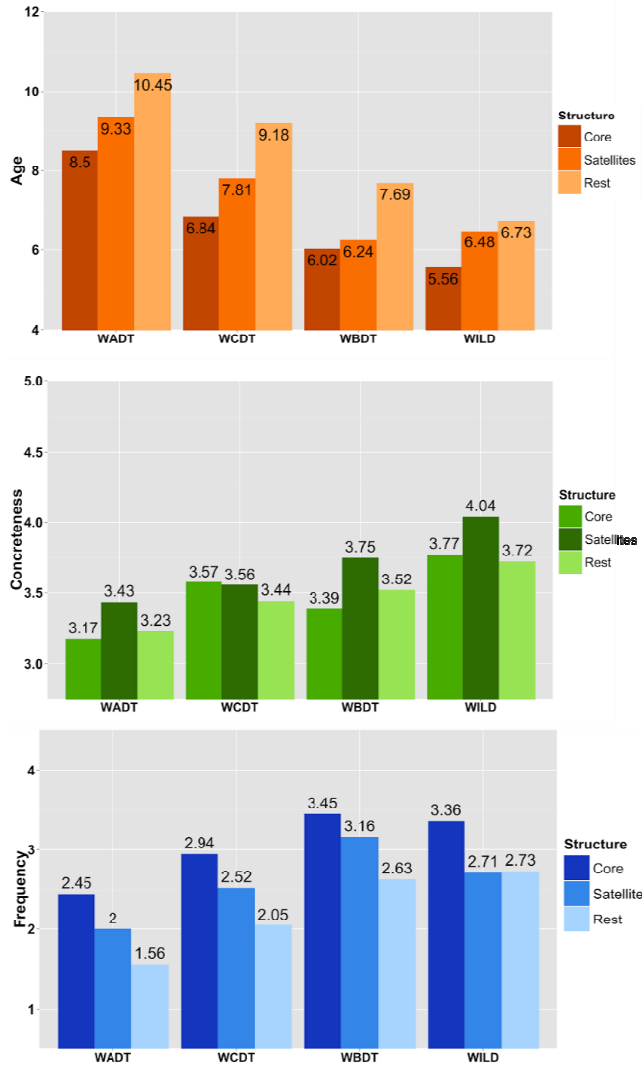


Figure 5.3. *Word frequency, age and concreteness in Core (C), Satellites (S) and rest of the Dictionary (D). Top: words in the Core are learned earlier than the Satellites, which are in turn learned earlier than the rest of the Dictionary. The pattern is the same in all four dictionaries. Middle: In three of the four dictionaries, the Satellite words are more concrete than the Core words and the rest of the Dictionary (WADT, WBDT, WILD); no pattern for concreteness in WCDT. Bottom: Core words are more frequent than the Satellites, which are more frequent than the rest of the Dictionary. The pattern is the same across the four dictionaries for word frequency except that, in the smallest, WILD, there is no significant difference between Satellites and the rest of the Dictionary. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

The Kernel contains a very large number of word combinations that can form a MinSet, but each dictionary's MinSet is composed of the same proportion of C and S words (from 57% C in the largest, WADT, to 88% C in the smallest, WILD). What is the complementary role of the Core and the Satellites in generating a MinSet?

To get a better idea of which Kernel words are included in MinSets, we compared their frequency, age and concreteness with those of randomly selected Kernel words. All MinSets are part-Core and part-Satellite. For a given dictionary, some Kernel words are in all MinSets, some in no MinSet, and the rest are in between. However, the number of MinSets for a given dictionary is too large to enumerate them all and calculate the percentage of MinSets in which each Satellite and Core word appears.[6] We therefore used a Monte Carlo simulation to compare each dictionary's real MinSets with randomly selected pseudo-MinSets of the same size and Core/Satellite ratio.

---

6 A graph D is a dictionary graph (or dictionary for short) if each of its nodes (words), w, has at least one predecessor (defining word), w'. The set of all words of D is denoted by W and its set of arcs (the directed definitional links from each defining word w' to each defined word w) is denoted by L.

A sub-dictionary of D is a subset D' of W with the property that for any word w in W, each predecessor w' of w in D also belongs to W. Mathematically speaking, (word w belongs to W) and (link(w',w) belongs to L) together imply that (w' belongs to W).

If W is a strongly connected component (SCC) of D, then no proper subset W' of W (i.e. no subset W' such that there is at least one word w belonging to W but not to W') is a sub-dictionary of D.

Hence, a sub-dictionary is always the union of a collection of SCCs: By contracting each SCC (i.e. replacing each SCC by one supernode), we obtain an acyclic graph. We can thus speak of the SCCs that precede a given SCC.

An acyclic graph is a graph with no cycles, i.e. no sequence of the form (w1, w2, w3,…, wn, w1), in which each word in the sequence is used in the definition of the word that follows it in the sequence.

The union of a collection of SCCs is a sub-dictionary if and only if each predecessor of an SCC in the collection also belongs to the collection.

To find a MinSet for the Core C, we first apply certain reductions (see [LEV 88, LIN 00] to obtain a smaller graph called the reduced Core and denoted by C'. Finding a MinSet for C' is less difficult than finding one for C itself, although it remains a non-trivial task. Once a MinSet has been found for C', it is easy to transform it into a MinSet for C.

For the C' of each of our dictionaries, we have been able to determine that they contain three classes of words: essential words (a word is essential if it belongs to every MinSet), superfluous words (a word is superfluous if it does not belong to any MinSet) and ordinary words (a word is ordinary if it is neither essential nor superfluous).

We still have to extend these results to the Core itself and to the full dictionary, but it is likely that we will find essential words and superfluous words for these sets as well.

We then computed the average value of each psycholinguistic variable for each part-core and part-satellite independently between the MinSets and the pseudo-MinSets (see Figure 5.4).

With a few exceptions, the same pattern was observed for all four dictionaries. The Core words in the real MinSets are younger, more concrete and more frequent than those in the random pseudo-MinSets. The Satellite words in the real MinSets are less young and less frequent than their pseudo-MinSet counterparts; no clear pattern was observed for concreteness. T-tests for all paired comparisons between the MinSets and random samples were all highly significant ($p < 0.01$) with the exception of one comparison (concreteness for the Satellite part of WBDT MinSets).

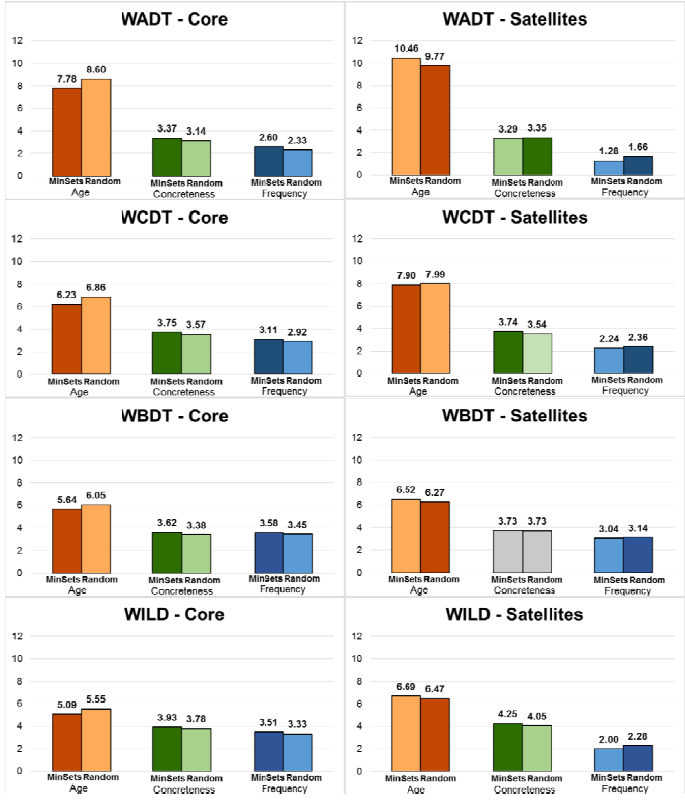


Figure 5.4. *Word frequency, age and concreteness within real MinSets versus random Pseudo-MinSets. Core/Satellite differences are magnified in MinSets compared with random samples of Core and Satellite words in the same proportion. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

5.4. Discussion

Our findings suggest that in addition to the overall tendency for words to be younger and more frequent as we move from the outer rest of the dictionary to the Satellites to the Core (an effect that is accentuated in the MinSets), something different is happening at the Satellite layer, which is more concrete than the deeper Core as well as the rest of the dictionary. The Satellite words are more concrete and they are needed in every MinSet. We still do not know what functional role is played by the S words and the C words in forming a MinSet. Nor do we know what the differences are between MinSets. We doubt that anyone ever learns one MinSet from direct experience and then learns everything else through verbal definition, but our findings do suggest that it is in the nature of lexical representation that this is possible in principle.

In this work, we extended our previous analyses to four new dictionaries not included in our prior studies. Our objective was twofold. First, we wanted to replicate our findings and show that the hidden structures we had discovered were present in any dictionary. Second, the three smaller dictionaries of the Wordsmyth suite are purpose-designed for learners at different levels, which makes them interesting from the standpoint of language acquisition. As we move from the advanced (WADT) to the simplest dictionary (WILD), the relative proportion of the Kernel increases markedly and so does the relative size of the structures within it (S, C, M). One conclusion from this might have been that smaller dictionaries simply have larger Kernels. However, this was not true in the dictionaries we had studied previously, where the Kernel remained between 8 and 12% of the dictionary despite huge variation in dictionary size [VIN 16]. One possible explanation is that it is precisely because the words in the three smaller Wordsmyth dictionaries with their larger Kernels were all deliberately selected in order to teach the English language that they contain a greater proportion of “grounding words”.

These results have implications for the understanding of symbol grounding and the learning and mental representation of word meaning. For

language users to be able to learn and understand the meaning of words from verbal definitions, they have to have the vocabulary to understand the words in those definitions, or at least to understand the definitions of the words in the definitions, and so on. They need an already grounded set of word meanings that is sufficient to carry them, verbally, to the meaning of any other word in the language, if they are to be able to learn its meaning through words alone. A Grounding Set clearly has to be acquired before it is true that all further words can be acquired verbally; hence, we would expect the Grounding Set to be acquired earlier in life. It also makes sense that the words in the Grounding Set are used more frequently, perhaps partly because they are used more often to define or explain later words.

Grounding words being more concrete is also to be expected, because word meanings that do not come from verbal definitions have to be acquired by nonverbal means, and those nonverbal means are likely to be the learning of categories through direct sensorimotor experience: learning what to do and not do with what kind of thing [HAR 10, PÉR 16]. It is easy, then, to associate a category that we have already learned non-verbally with the (arbitrary) name that a language community agrees to call it [BLO 13]. The words denoting sensorimotor categories are hence likely to be more concrete.

Categorization itself, however, is by its nature also abstraction: to abstract is to single out some features of a thing, and ignore others. The way we learn what *kinds* of things there are, and what to do and not do with them (including what to call them), is not by simply memorizing raw sensorimotor experiences by rote. To be able to do the right thing with the right kind of thing, we learn through trial-and-error sensorimotor interactions to detect and abstract the features that distinguish the members of a category from the non-members and to ignore the rest of the features as irrelevant. The process of abstraction in the service of categorization leads in turn to higher-order categories, which are hence more likely to be verbal categories rather than purely sensorimotor ones. For example, we can have a preverbal category for “bananas” and “apples”, based on their sensory projections and the differing sensorimotor actions needed to eat them; but the higher-order category

"fruit" is not as evident at a non-verbal level, being more abstract. It is also likely that having abstracted the sensorimotor features that distinguish the members and non-members of a concrete category nonverbally, we will not just give the members of the category a name, but we may go on to abstract and name their features (yellow, red, round, elongated) too. It may be that some names for more abstract, higher-order categories are as essential in forming a Grounding Set as the more concrete categories and their names and that this may have something to do with the complementary functional role played by the Satellites in the make-up of a MinSet.

Finally, the lexicon of the language – our repertoire of categories – is open-ended and always growing. To understand the grounding of meaning, it will be necessary not only to look at the growth across time of the vocabulary (both receptive and productive) of the child, adolescent and adult, but also the growth across time of the vocabulary of the language itself (diachronic linguistics), to understand which words are necessary, and when, in order to have the full lexical power to define all the rest [LEV 12]. We have discussed Minimal Grounding Sets (MinSets), and it is clear that there are potentially very many of these; but it is not clear that anyone uses just one MinSet, or could actually manage to learn everything verbally knowing just one MinSet. It is almost certain that we need some redundancy in our Grounding Sets. The Kernel, after all, is only five times as big as a MinSet. Perhaps we do not even need a full Grounding Set in order to get by, verbally; maybe we can manage with gaps, (certainly, the child must, at least initially.) Nor is it clear – even if we have full mastery of enough MinSets or even a Kernel – that the best way to learn the meaning of all subsequent words is from verbal definitions alone. Language may well have evolved in order to make something like that possible in principle: to acquire new categories through recombinatory subject/predicate propositions, purely by verbal "telling", without sensorimotor "showing" [BLO 13]. However, in practice, the learning of new word meanings may still draw on some hybrid show-and-telling.

5.4.1. *Limitations*

Many approximations and simplifications have to be taken into account in interpreting these findings. We are treating a definition as an unordered string of words, excluding functional (stop) words and not making use of any syntactic structure. Many words have multiple meanings, and we are using only the first meaning of each word. The problem of extracting MinSets is NP-hard. In the special case of dictionary graphs -- and thanks also to the empirical fact that the Core turns out to be so big, and surrounded by small Satellites -- we have been able, using the algorithm of Lin and Jou [LIN 00] and techniques from integer linear programming (e.g. [NEM 99]), to extract a number of MinSets for the Wordsmyth suite of dictionaries, whose results we are reporting here as well as for other English dictionaries [VIN 16], such as Merriam-Webster and WordNet [FEL 10]. The analysis is now being extended to dictionaries in other languages.

5.5. Future work

In order to compare the emerging hidden structure of dictionaries with the way word meaning is represented in the mind (the "mental lexicon"), we have also created an online dictionary game in which the player is given a word to define; they must then define the words they used to define the word, and so on, until they have defined all the words they have used. This generates a mini-dictionary of a much more tractable size (usually less than 200 words; Figures 5.5 and 5.6)[7]. http://lexis.uqam.ca:8080/dictGame

We are currently performing the same analyses on these much smaller mini-dictionaries, to derive their Kernel, Core, Satellites and MinSets, and

7 The 37-word mini-dictionary in Figures 5.5 and 5.6 is displayed because it is small enough to illustrate the hidden dictionary graph structure at a glance. It was generated before we had added a new rule that a definition is not allowed to be just a synonym: in the more recent version of the game, a definition has to be at least two content words (and we may eventually also rule out second-order circularity [A = B + C, B = C + not-A, C = A + not-B]. However, it has to be borne in mind that (because of the symbol grounding problem) every dictionary is necessarily approximate and (at some level) circular (much the way all SCCs [other than trivial single-word SCCs] are circular). This is true whether it is a full dictionary or a game mini-dictionary generated by one player. Definitions can only convey new meanings if the mind already has enough old meanings, grounded by some means other than definition.

their psycholinguistic correlates (age, concreteness, frequency), to determine whether these inner "mental" dictionaries share the hidden structure and function that we are discovering in the formal external lexicon (see Figures 5.5 and 5.6). These mini-dictionaries will also allow us to analyze the difference in the functional role between the Satellite words and the Core words that make up a MinSet, which is much more difficult to do with full-size dictionaries.

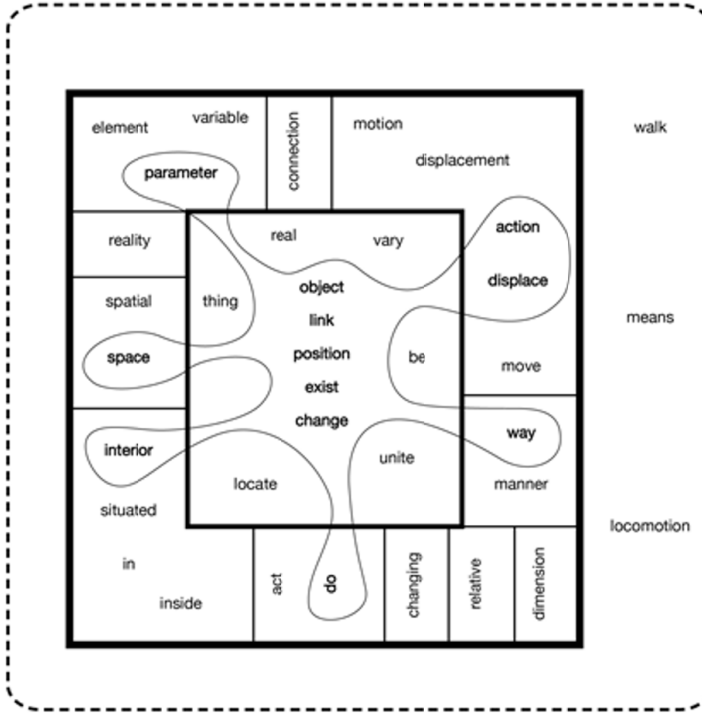


Figure 5.5. *Mini-dictionary diagram. The diagram is the same as Figure 5.1, but with real words to provide a concrete example. This 37-word mini-dictionary was generated by a player of our online dictionary game. The player is given a word and must define that word, as well as all the words used to define it, and so on, until all the words used are defined. The smallest resulting dictionary so far (37 words) is used here to illustrate the mini-dictionary's Kernel and Core plus one of its MinSets. Note that all the words in this mini-dictionary are in the Kernel except the start word, "walk", plus "locomotion" and "means". Figure 5.6 displays the graph for this mini-dictionary*

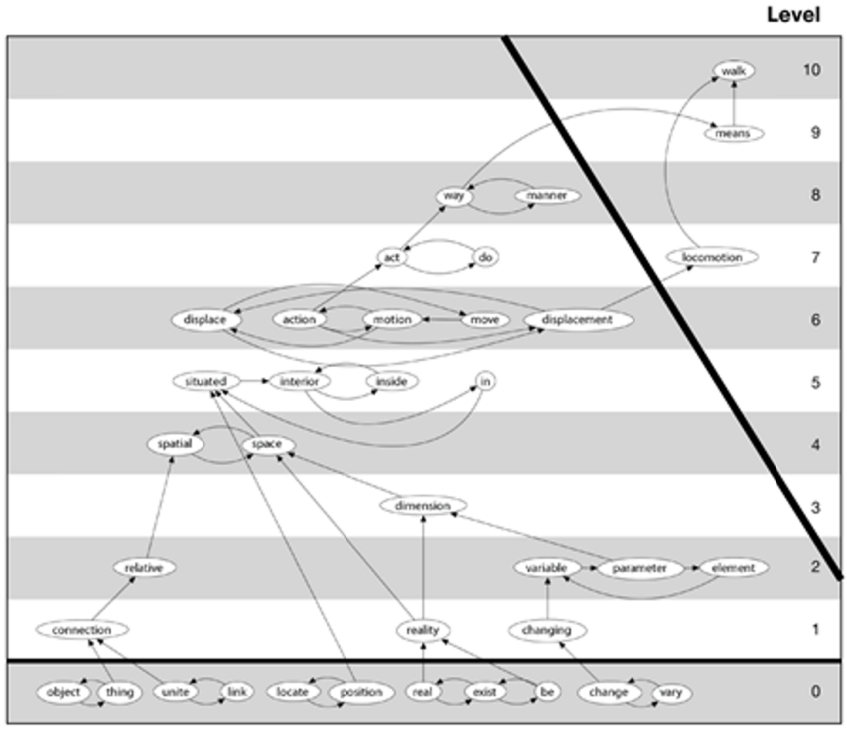


Figure 5.6. *Mini-dictionary graph. Graph of mini-dictionary in Figure 5.5, showing the definitional links. Note that, in this especially tiny mini-dictionary, unlike in the full dictionaries and many of the other mini-dictionaries, the words in the Core (level 0), rather than being the single largest SCC, are the union of multiple SCCs. The oblique boldface line separates the Kernel from the (three) words in the rest of this mini-dictionary*

5.6. Bibliography

[BLO 08] BLONDIN-MASSÉ A., CHICOISNE G., GARGOURI Y. *et al.*, "How is meaning grounded in dictionary definitions?" *TextGraphs-3 Workshop – 22nd International Conference on Computational Linguistics*, available at: http://www.archipel.uqam.ca/657/, 2008.

[BLO 13] BLONDIN-MASSÉ A., HARNAD S., PICARD, O. *et al.*, "Symbol grounding and the origin of language: from show to tell", in LEFEBVRE C., COMRIE B., COHEN H. (eds), *Current Perspective on the Origins of Language*, John Benjamins Publishing Company, Amsterdam, 2013.

[BRY 09] BRYSBAERT M., NEW B., "Moving beyond Kučera and Francis: a critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English", *Behavior Research Methods*, vol. 41, no. 4, pp. 977–990, 2009.

[BRY 14] BRYSBAERT M., WARRINER, A.B., KUPERMAN V., "Concreteness ratings for 40 thousand generally known English word lemmas", *Behavior Research Methods*, vol. 46, no. 3, pp. 904–911, 2014.

[FEL 10] FELLBAUM C., *WordNet*, Springer, Netherlands, 2010.

[FOM 08] FOMIN F.V., GASPERS S., PYATKIN A.V. *et al.*, "On the minimum feedback vertex set problem: exact and enumeration algorithms", *Algorithmica*, vol. 52, no. 2, pp. 293–307, 2008.

[HAR 90] HARNAD S., "The symbol grounding problem", *Physica D*, vol. 42, pp. 335–346, 1990.

[HAR 05] HARNAD S., "To cognize is to categorize: cognition is categorization", in LEFEBVRE C., COHEN H. (eds), *Handbook of Categorization*, Elsevier, Amsterdam, 2005.

[HAR 10] HARNAD S., "From sensorimotor categories and pantomime to grounded symbols and propositions", in TALLERMAN M., GIBSON K.R. (eds), *The Oxford Handbook of Language Evolution*, Oxford University Press, Oxford, 2010.

[KAR 72] KARP R.M., "Reducibility among combinatorial problems", in MILLER R.E., THATCHER J.W., BOHLINGER J.D. (eds), *Proceedings of a Symposium on the Complexity of Computer Computations*, IBM Thomas J. Watson Research Center, New York, 1972.

[KUP 12] KUPERMAN V., STADTHAGEN-GONZALEZ H., BRYSBAERT M., "Age-of-acquisition ratings for 30,000 English words", *Behavior Research Methods*, vol. 44, no. 4, pp. 978–990, 2012.

[LAP 12] LAPOINTE M., BLONDIN-MASSÉ A, GALINIER P. *et al.*, "Enumerating minimum feedback vertex sets in directed graphs", *Bordeaux Graph Workshop*, Bordeaux, 2012.

[LEV 12] LEVARY D., ECKMANN J.P., MOSES E. *et al.*, "Loops and self-reference in the construction of dictionaries", *Physical Review X*, vol. 2, no. 3, pp. 031018, 2012.

[LEV 88] LEVY H., LOW, D.W., "A contraction algorithm for finding small cycle cutsets", *Journal of Algorithms*, vol. 9, no. 4, pp. 470–493, 1988.

[LIN 00] LIN H.M., JOU J.Y., "On computing the minimum feedback vertex set of a directed graph by contraction operations", *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 19, no. 3, pp. 295–307, 2000.

[NEM 99] NEMHAUSER G., WOLSEY L., *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1999.

[PAR 98] PARKS R., RAY J., BLAND S., *Wordsmyth English Dictionary/Thesaurus [WEDT]*, University of Chicago, Chicago, 1998.

[PÉR 16] PÉREZ-GAY F, SABRI H., RIVAS D. *et al.*, "Perceptual changes induced by category learning", *23rd Annual Meeting Cognitive Neuroscience Society*, New York, April 2016.

[VIN 16] VINCENT-LAMARRE P., BLONDIN M.A., LOPES M. *et al.*, "The latent structure of dictionaries", *Topics in Cognitive Science*, 2016.

6

Transductive Learning Games for Word Sense Disambiguation

This chapter presents a semi-supervised approach to word sense disambiguation, formulated in terms of evolutionary game theory, where each word to be disambiguated is represented as a player and each sense as a strategy. The players receive a reward for interacting with other players, which gives them an incentive to select strategies with a higher payoff. The interactions among players are modeled with a weighted graph and it is assumed that some players have a defined strategy (labeled players) and others have to choose their strategy (unlabeled players). The information is propagated over the graph from labeled players to unlabeled, exploiting information from different sources: word similarity that weights the importance of the interactions and sense similarity that determines the payoffs of the games. In this way, similar players influence each other, selecting correlated senses. The method has been tested on four datasets with different numbers of labeled words. The experimental results demonstrate that the proposed approach performs well compared with state-of-the-art algorithms.

6.1. Introduction

Word Sense Disambiguation (WSD) is the task of identifying the intended sense of a word in a computational manner based on the context in which it appears [NAV 09]. Understanding the ambiguity of natural languages is considered an AI-hard problem [MAL 88]. Computational problems like this are the central objectives of Artificial Intelligence (AI) and Natural Language Processing (NLP) because they aim to solve the epistemological question of how the mind works. It has been studied since the beginning of NLP [WEA 55], and today is a central topic of this discipline.

Chapter written by Rocco TRIPODI and Marcello PELILLO.

The identification of the intended meaning of the words in a sentence is a hard task and humans have problems with it. This happens because people use words in a different manner from their literal meaning, and misinterpretation is a consequence of this habit. To solve this task, it is not only required to have a deep knowledge of the language, but also to be an active speaker of the language. In addition, languages change over time, in an evolutionary process, by virtue of the use of language by speakers, which leads to the formation of new words and new meanings which could be understood only by active speakers of the language. In fact, active speakers can identify the intended meanings of the words according to the knowledge of the communication context in which the words are expressed.

WSD is also a central topic in applications such as text entailment [DAG 04], machine translation [VIC 05], opinion mining [SMR 06] and sentiment analysis [REN 09]. These applications require the disambiguation of ambiguous words, as preliminary process; otherwise, they remain on the surface of the word [PAN 02], compromising the coherence of the data to be analyzed.

Our approach to WSD is a graph based on the way the data are represented, similarity based on the way the senses and the words of a sentence are compared, and is formulated in game theoretic terms to combine this information and to find consistent labeling of the data. It is aimed at maximizing the textual coherence imposing that the meaning of each word in a text must be related to the meaning of the other words in it. To do this, we exploited distributional and proximity information to weight the influence that each word has on the others. We also exploited semantic similarity information to weight the strengths of compatibility among senses and use transductive learning principles to propagate the information over the graph.

The rest of this chapter is structured as follows: section 6.2 presents graph-based algorithms for WSD. Section 6.3 introduces our approach to semi-supervised learning and some notions of game theory. This chapter continues with the presentation of our methods in section 6.4 and concludes with section 6.5 where we present the evaluation of our system and the comparison of our approach with state-of-the-art algorithms.

6.2. Graph-based word sense disambiguation

Graph-based WSD algorithms try to identify the actual sense of a word in a determined phrase, exploiting the information derived from its context. They are gaining much attention in the NLP community. This is because graph theory is a powerful tool that can be employed for different purposes, from the organization of the contextual information to the computation of the relations among word senses. These kind of approaches construct a graph collecting all the possible senses of the words in a text and represent them as nodes. Then, using connectivity measures can identify the most relevant word senses in the graph [SIN 07, NAV 07a].

Navigli and Lapata [NAV 07a] conducted an extensive analysis of graph connectivity measures for unsupervised WSD. Their approach uses a knowledge base, such as WordNet, to collect and organize all the possible senses of the words to be disambiguated in a graph structure, then uses the knowledge base to search for a path between each pair of senses in the graph and if it exists, it adds all the nodes and edges on this path to the graph. These measures analyze local and global properties of the graph. The results of the study indicate that local measures outperform global measure and, in particular, degree centrality and PageRank [PAG 99] achieve the best results.

PageRank [PAG 99] is one of the most popular algorithms for WSD; in fact, it has been implemented in several different ways by the research community [MIH 04, HAV 02, AGI 14, DEC 10]. It represents the senses of the words in a text as nodes of a graph. It uses a knowledge base to collect the senses of the words in a text and represents them as nodes of a graph. The structure of the knowledge base is used to connect each node with its related senses in a directed graph. The main idea of this algorithm is that whenever a link from one node to another node exists, a vote is produced, increasing the rank of the voted node. It works by counting the number and quality of links to a node to determine an estimation of how important the node is in the network. The underlying assumption is that more important nodes are likely to receive more links from other nodes [PAG 99]. Exploiting this idea, the ranking of the nodes in the graph can be computed iteratively with the following equation:

$$Pr = c\, M\, Pr + (1 - c)\, v$$

where M is the transition matrix of the graph, v is a $N \times 1$ vector representing a probability distribution and c is the so-called damping factor that represents the chance that the process stops, restarting from a random node. At the end of the process, each word is associated with the most important concept related to it. One problem of this framework is that the labeling process is not assumed to be consistent.

An algorithm, which tries to improve centrality algorithms, is SUDOKU, introduced by Manion and Sainudiin [MAN 14]. It is an iterative approach that simultaneously constructs the graph and disambiguates the words using a centrality function. It starts inserting in the graph the nodes corresponding to the senses of the words with low polysemy. The advantages of this method are that it uses small graphs at the beginning of the process, reducing the complexity of the problem; furthermore, it can be used with different centrality measures.

Recently, a model based on an undirected graphical model has been introduced by [CHA 15]. This method interprets the WSD problem as a maximum a posteriori query on a Markov random field [JOR 02]. The graph is constructed using the content words of a sentence as nodes and connecting them with edges if they share a relation, determined using a dependency parser. The values that each node in the graphical model can take include the senses of the corresponding word. The senses are collected using a knowledge base and weighted using a probability distribution based on the frequency of the senses in the knowledge base. Furthermore, the senses between two related words are weighted using a similarity measure. The goal of this approach is to maximize the joint probability of the senses of all the words in the sentence, given the dependency structure of the sentence, the frequency of the senses and the similarity among them.

A new graph-based, semi-supervised approach, introduced to deal with multilingual WSD [NAV 12b] and entity linking problems, is Babelfy [MOR 14]. Multilingual WSD is an important task because traditional WSD algorithms and resources are mainly focused on English language. It exploits the information from large multilingual knowledge bases, such as BabelNet [NAV 12a] to perform this task. Babelfy creates the semantic signature of each word to be disambiguated, that consists of collecting, from a semantic network, all the nodes related to a particular concept, exploiting the global structure of the network. This process leads to the construction of a graph-based representation of the whole text. Then, it applies Random Walk with

Restart [TON 06] to find the most important nodes in the network, solving the WSD problem.

Approaches that are more similar to ours in the formulation of the problem have been described by Araujo [ARA 07] and a specific evolutionary approach to WSD has been introduced by Menai [MEN 14]. It uses genetic algorithms [HOL 75] and memetic algorithms [MOS 89] to improve the performances of a gloss-based method. It assumes that there is a population of individuals, represented by all the senses of the words to be disambiguated and that there is a selection process that chooses the best candidates in the population. The selection process is defined as a sense similarity function that gives a higher score to candidates with specific features, increasing their fitness. This process is repeated until the fitness level of the population regularizes and, at the end, the candidates with higher fitness are selected as solutions of the problem.

6.3. Our approach to semi-supervised learning

Our approach uses some labeled nodes to propagate the information over the graph disambiguating unlabeled nodes in a consistent way. This process is based on two fundamental principles: the homophily principle, borrowed from social network analysis, and the transductive learning. The former simply states that similar objects are expected to have the same class [EAS 10]. We extended this principle assuming that objects, which are similar, are expected to have a similar class; an idea used also by [KLE 02], within a Markov random field framework. The latter is a case of semi-supervised learning [SAM 11] particularly suitable for relational data (see section 6.3.1).

In our system, we used a graph to model the geometry of the data and an evolutionary process to propagate the information over it. The graph construction method is described in section 6.4.1 and the evolutionary process in section 6.4.4. This work extends our previous works on unsupervised and semi-supervised WSD [TRI 15a, TRI 17].

6.3.1. *Graph-based semi-supervised learning*

Transductive learning was introduced by Vladimir Vapnik [VAP 98]. It was motivated by the fact that it is easier than inductive learning, given the

fact that inductive learning tries to learn a general function to solve a specific problem, while transductive learning tries to learn a specific function for the problem at hand.

It consists of a set of labeled objects $(x_i, y_i)\,(i = 1, 2, ..., 1)$, where $x_i \in \mathbb{R}^n$ are objects represented by real-valued attributes and $y_i \in (1, 2, ..., m)$ are the possible labels of these objects. Together with the labeled objects, there is also a set of k unlabeled objects $(x_{l+1},\ldots,x_{l+k})$. Rather than finding a general rule for classifying future examples, transductive learning aims at classifying only (the k) unlabeled objects exploiting the information derived from labeled ones.

Within this framework, it is common to represent the geometry of the data as a weighted graph. For a detailed description of algorithms and applications on this field of research, named graph transduction, we refer to [ZHU 05]. The purpose of this method is to transfer the information given by labeled nodes to unlabeled ones, exploiting the graph structure. Formally, we have a graph $G = (V, E, w)$ in which V is the set of nodes representing both labeled and unlabeled points, $V = \{v_l, v_u\}$, E is the set of edges $E \subseteq V \times V$ connecting the nodes of the graph and $w: \varepsilon \to \mathbb{R}^+$ is a weight function assigning a similarity value to each edge $\varepsilon \in E$. The task of transduction learning is to estimate the labels of the unlabeled points, given the pairwise similarity among the data points and a set of possible labels $\varphi = \{1, ..., c\}$.

6.3.2. *Game theory and game dynamics*

Game theory was introduced by Von Neumann and Morgenstern [VON 44] in order to model the essentials of decision-making in interactive situations. In its normal-form representation, it consists of a finite set of players $I = \{1, .., n\}$, a set of pure strategies for each player $S_i = \{s_1,\ldots,s_m\}$ and a utility function $S_1 \times \ldots \times S_n \to \mathbb{R}$, which associates strategies with payoffs. Each player can adopt a strategy in order to play a game and the obtained payoff depends on the combination of strategies played at the same time by two players (strategy profile). In non-cooperative games, the players

choose their strategies independently, considering what other players can play and trying to find the best strategy to employ in a game.

The players can play mixed strategies, which are probability distributions over pure strategies. A mixed strategy can be defined as a vector $x = \{x_1, \ldots, x_m\}$, where m is the number of pure strategies and each component x^h denotes the probability that a player chooses its h-th pure strategy. Each mixed strategy corresponds to a point on the simplex, whose corners correspond to pure strategies.

A strategy profile can be defined as a pair (p,q), where $p \in \Delta_i$ and $q \in \Delta_j$. The expected payoff for this strategy profile is computed as:

$$u_i(p,q) = p \cdot A_i q$$

and

$$u_j(p,q) = q \cdot A_j p$$

where A_i and A_j are the payoff matrices of players i and j respectively.

In evolutionary game theory, we have a population of agents that play games repeatedly with their neighbors and update their beliefs on the state of the system, choosing their strategy according to what action has been effective and what has not in previous games, until the system converges. The strategy space of each player i is defined as a mixed strategy x_i, as defined above. The payoff corresponding to a single strategy can be computed as:

$$u(x_i^h) = \sum_{j=1}^{n} (A_{ij} x_j)^h$$

and the average payoff is:

$$u(x_i) = \sum_{j=1}^{n} x_i^T A_{ij} x_j$$

where n is the number of players with whom the games are played and A_{ij} is the payoff matrix among players i and j. The replicator dynamic equation [TAY 78] is used to find those states of the system that correspond to the Nash equilibria of the games:

$$x_i^h(t+1) = x_i^h(t)\frac{u\left(x_i^h(t)\right)}{u\left(x_i(t)\right)} \forall h \in x_i$$

This equation allows better than average strategies to grow at each iteration. Each iteration of the dynamics can be considered as an instance of an inductive learning process, in which the players learn from the others how to play their best strategy in a determined context. When equilibrium is reached, we assign to each player the label corresponding to the strategy with the highest value, which is computed with the following equation:

$$\phi_i = \underset{h=1,\ldots,m}{\operatorname{argmax}} x_i^h .$$

Experimentally, we noticed that the selected values are always close to 1.

6.4. Word sense disambiguation games

In this section, we describe how we formulate the WSD problem in game theoretic terms, extending our previous works [TRI 15a, TRI 15b, TRI 17] with the use of transductive learning principles. In the next section, we describe how the interaction graph is constructed, in section 6.4.2 we describe how the strategy space of the players is initialized, then we introduce the payoff matrices of the games and the system dynamics.

6.4.1. *Graph construction*

The graph is constructed selecting from a text all the words that have an entry in a knowledge base such as WordNet [FEL 98], denoted by $I = \{1,\ldots,N\}$, where N is the number of target words. From I, we

constructed the $N \times N$ similarity matrix W where each element w_{ij} is the similarity among words i and j. W can be exploited as a useful tool for graph-based algorithms, since it is treatable as a weighted adjacency matrix of a weighted graph.

A crucial factor for the graph construction is the choice of the similarity measure, $sim(\cdot,\cdot) \rightarrow \mathbb{R}$ to weight the edges of the graph. For our experiments, we used the Dice coefficient [DIC 45], since it has performed well on different datasets [TRI 15a, TRI 17]. This measure determines the strength of co-occurrence between two words and is computed as follows:

$$Dice(i,j) = \frac{2c(i,j)}{c(i)+c(j)}$$

where $c(i)$ is the total number of occurrences of word i in a large corpus and $c(i,j)$ is the co-occurrence of the words i and j in the same corpus. This formulation is particularly useful to decrease the ranking of words that tend to co-occur frequently with many other words. For the experiments in this work, we used as corpus the British National Corpus [LEE 92]. The similarity graph W encodes the information of how two target words are similar, in a distributional semantics perspective [HAR 54].

6.4.2. *Strategy space*

The strategy space of the players is created using a knowledge base to collect the sense inventories $S_i = \{1,\ldots,m_i\}$ of each word in the text, where m_i is the number of senses associated with word i. Then, it creates the list $C = (1,\ldots,c)$ of the unique senses in all inventories, which corresponds to the space of the game.

With this information, it is possible to initialize the mixed strategy space x of each player. It can be initialized using a uniform distribution or considering information from sense labeled corpora, allocating more mass

to frequent senses. In the former case, we initialize the strategy spaces of each player with the following equation:

$$x_i^h = \begin{cases} m_i^{-1}, \text{ if sense } h \text{ is in } S_i \\ 0, \text{ otherwise} \end{cases}, \forall h \in C$$

In the latter case, we assign to each sense a probability according to its rank, assigning higher probabilities to senses with a high frequency. To model this scenario, we used a geometric distribution that produces a decreasing probability distribution. This initialization is defined as follows:

$$x_i^h = \begin{cases} p(1-p)^{r^h}, \text{ if sense } h \text{ is in } S_i \\ 0, \text{ otherwise} \end{cases}, \forall h \in C,$$

where p is the parameter of the geometric distribution and determines the scale or statistical dispersion of the probability distribution, and r^h is the rank of sense h, which ranges from 1, the rank of the most common sense for word i, to m_i, the rank of the least frequent sense. These values are divided by $\sum_{h \in C} x^h$ to make them add up to 1. In our experiments, we used the ranked system provided by the Natural Language Toolkit (version 3.0) [BIR 06] to rank the senses associated with each word to be disambiguated.

6.4.3. *The payoff matrix*

We encoded the payoff matrix of the games as a sense similarity matrix among all the senses in the strategy spaces of the game. In this way, the higher the similarity between the senses of two words, the higher the incentive for a player to select that sense, and play the strategy associated with it.

The $c \times c$ sense similarity matrix A is defined in the following equation:

$$a_{ij} = sim\left(a_i, a_j\right) \forall i, j \in C : i \neq j$$

This matrix can be computed using the information derived from the same knowledge base used to construct the strategy space of the game. It is used to extract the partial payoff matrix A_{ij} for all the single games played

between two players i and j. This operation is performed by extracting from A the entries relative to the indices of the senses in the sense inventories S_i and S_j. It produces an $m_i \times m_j$ payoff matrix, where m_i and m_j are the numbers of senses in S_i and S_j, respectively.

The semantic measure that we used in this work is the Gloss Vector measure [PAT 06], since it has been demonstrated to have stable performances in different datasets [TRI 17]. It is based on the computation of the similarity between the definitions of two concepts in a lexical database. They are used to construct a co-occurrence vector $v_i = (v^1, v^2, \ldots, v^n)$ for each concept i, with a bag-of-words approach, where v^h represents the number of times word v occurs in the gloss and n is the total number of different words in the corpus. From this representation, it is possible to compute the similarity between two vectors using the cosine distance:

$$cos\theta \frac{v_i . v_j}{||v_i|| |v_j||}$$

The vectors are constructed using the concept of super-gloss introduced by [PAT 06]. It is the concatenation of the gloss of the synset and the glosses of the synsets connected to it with any relation in the knowledge base.

6.4.4. *System dynamics*

At each iteration of the system, each player plays a game with its neighbors N_i according to the graph W. The payoff of the h-th strategy is calculated as:

$$u\left(x^h\right) = \sum_{j \in N_i} (w_{ij} A_{ij} x_j)^h$$

and the player's payoff as:

$$u(x) = \sum_{j \in N_i} x_i^T (w_{ij} A_{ij} x_j)$$

where N_i represents the neighbors of player i. We assume that the payoff of word i depends on the similarity that it has with word j, w_{ij}, the similarities among its senses and those of word j, A_{ij}, and the sense preference of word j, (x_j), that can be unambiguous if j is a labeled player.

We use the replicator dynamics equation (see section 6.3.2) to find the Nash equilibria of the games. During each phase of the dynamics, a process of selection allows strategies with a higher payoff to emerge and at the end of the process each player chooses its sense according to these constraints.

6.5. Evaluation

In this section, we present the experimental setting for the evaluation and comparison of our system with state-of-the-art algorithms.

6.5.1. *Experimental setting*

We evaluated our algorithm with three fine-grained datasets: Senseval-2 English all-words[1] (S2) [PAL 01], Senseval-3 English all-words[2] (S3) [SNY 04], SemEval-2007 all-words[3] (S7) [PRA 07] and one coarse-grained dataset, SemEval-2007 English all-words[4] (S7CG) [NAV 07b], using WordNet as a knowledge base. The descriptions of the datasets are presented in Table 6.1.

The results of the evaluation are presented as $F1$, which is calculated as:

$$F1 = 2 \cdot \frac{precision\ recall}{precision + recall}.$$

This measure determines the weighted harmonic mean of precision and recall. Precision is defined as the number of correct answers divided by the number of provided answers and recall is defined as the number of correct

1 www.hipposmond.com/senseval2
2 http://www.senseval.org/senseval3
3 http://nlp.cs.swarthmore.edu/semeval/tasks/index.php
4 http://lcl.uniroma1.it/coarse-grained-aw

answers divided by the total number of answers to be provided. In our evaluation, we excluded labeled points in this calculation. Experimentally we noticed that precision is always equal to recall, since the system is always able to provide an answer.

We evaluated two different versions of the system, one using a uniform probability distribution to initialize the strategy space of the games and the other using information from sense labeled corpora (see section 6.4.2). Furthermore, to make the evaluation unbiased, we present the mean and standard deviation results of our system over 25 trials with different sizes of randomly selected labeled points.

Dataset	Text	N	C	Tot. N
S2	1	670	2195	2387
S2	2	997	1836	
S2	3	720	1916	
S3	1	783	2472	2007
S3	2	633	1426	
S3	3	591	1881	
S7	1	111	593	455
S7	2	150	798	
S7	3	194	1035	
S7CG	1	368	1287	2268
S7CG	2	379	1473	
S7CG	3	499	1926	
S7CG	4	677	1666	
S7CG	5	345	1410	

Table 6.1. *Number of target words and senses for each text of the datasets*

6.5.2. *Evaluation results*

The results of the evaluation are presented in Figures 6.1 and 6.2, where the results with the two initializations described in section 6.4.2 are shown: uniform (Figure 6.1) and geometric (Figure 6.2).

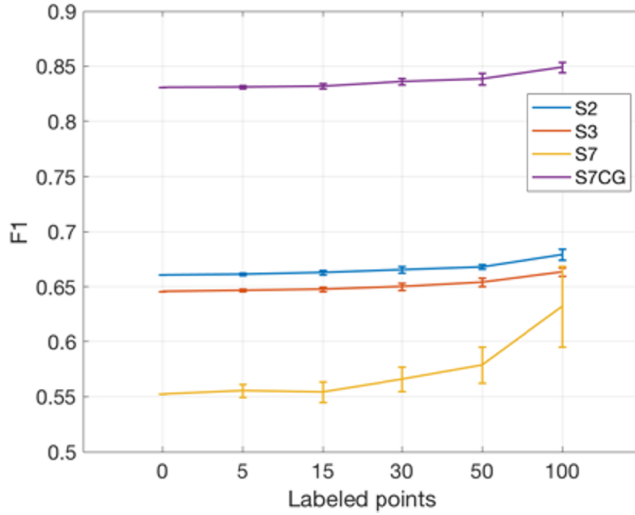


Figure 6.1. *Uniform distribution. Results as F1 varying the number of labeled nodes. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

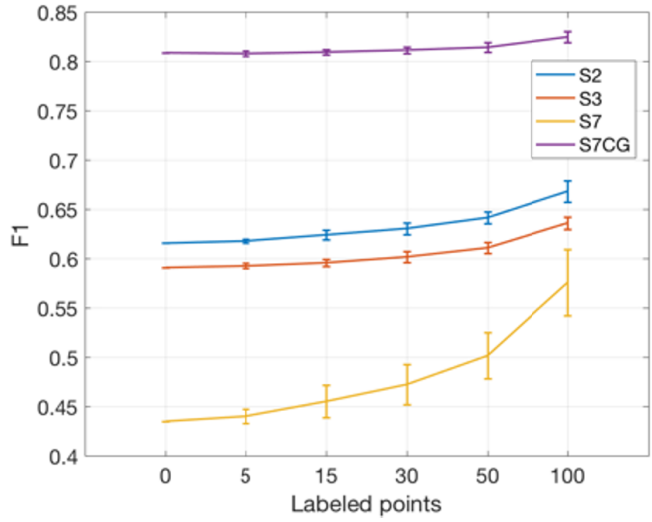


Figure 6.2. *Geometric distribution. Results as F1 varying the number of labeled nodes. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

As we can see from the plots, the performance of our system on S7CG is very different from the others. This is because this dataset is coarse grained which means that the disambiguation of each word is not restricted to just one sense, as in the fine-grained datasets, but to a set of similar senses.

An important aspect to note is that the performance of the system is always increasing with the increasing labeled points. This is particularly evident on S7, where the performance passes from 0.43 to 0.57 using the uniform distribution and from 0.55 to 0.63 using the geometric distribution. For the other datasets, the improvements given by the labeled point are in the range of $3-5\%$.

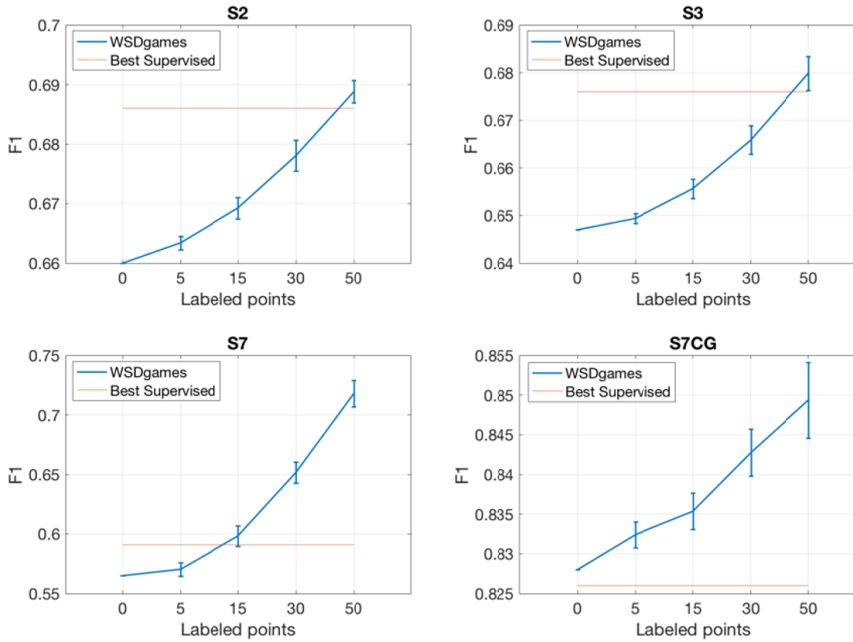


Figure 6.3. *Results as F1 using the geometric distribution and considering as correct the labeled nodes. The results are compared with the best supervised system on each dataset. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

The information given by labeled points is more effective when we use a uniform distribution to initialize the strategy space of the system. This can be explained considering that with this initialization, we use less information, and for this reason, the presence of labeled points can balance this lack.

6.5.3. *Comparison with state-of-the-art algorithms*

The comparison of our system using a geometric distribution to initialize the strategy space of the games is presented in Figure 6.3. We compared our results with the best system that participated in each competition on each dataset if their performances are higher than those obtained with *It makes sense*[5] [ZHO 10], a well-known supervised system.

From the plots, we can see that, on S7CG, the performance of our system is higher than those of supervised systems without using labeled points. This setting is the same as the one proposed in [TRI 17]. On the other datasets, we can see that the performance of our system follows a similar trend. In fact, on S2 and S3, we require 50 points to outperform supervised systems and, on S7, 15. These numbers correspond to the 2.09, 249 and 3.29 percent of S2, S3 and S7, respectively.

6.6. Conclusion

In this work, we presented a graph-based semi-supervised system for WSD, based on game theory and consistent labeling principles. Experimental results showed that our method improves the performance of conventional methods and that it requires a small amount of labeled points to outperform supervised systems. These systems require large corpora to be trained. These resources are difficult to create and are not suitable for domain specific tasks. Our system infers the meaning of a target word from a small amount of labeled data exploiting relational and contextual information. In fact, the information of a labeled point is not only used locally by near words but also propagated over the graph and used globally by the dynamical system obtained with our game theoretic framework.

5 This system achieves higher results on S7CG and S3.

6.7. Bibliography

[AGI 14] AGIRRE E., DE LACALLE O.L., SOROA A., "Random walks for knowledge-based word sense disambiguation", *Computational Linguistics*, vol. 40, no. 1, pp. 57–84, 2014.

[ARA 07] ARAUJO L., "How evolutionary algorithms are applied to statistical natural language processing", *Artificial Intelligence Review*, vol. 28, no. 4, pp. 275–303, 2007.

[BIR 06] BIRD S., "NLTK: the natural language toolkit", *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pp. 69–72, 2006.

[CHA 15] CHAPLOT D.S., BHATTACHARYYA P., PARANJAPE A., "Unsupervised word sense disambiguation using Markov random field and dependency parser", *AAAI*, pp. 2217–2223, 2015.

[DAG 04] DAGAN I., GLICKMAN O., "Probabilistic textual entailment: generic applied modeling of language variability", *Proceeding of Learning Methods for Text Understanding and Mining*, pp. 26–29, 2004.

[DEC 10] DE CAO D., BASILI R., LUCIANI M. *et al.*, "Robust and efficient page rank for word sense disambiguation", *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, pp. 24–32, 2010.

[DIC 45] DICE L.R., "Measures of the amount of ecologic association between species", *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[EAS 10] EASLEY D., KLEINBERG J., *Networks, Crowds, and Markets*, Cambridge University Press, Cambridge, 2010.

[FEL 98] FELLBAUM C., *WordNet*, Wiley Online Library, 1998.

[HAR 54] HARRIS Z.S., "Distributional structure", *Word*, vol. 10, nos. 2–3, pp. 146–162, 1954.

[HAV 02] HAVELIWALA T.H., "Topic-sensitive PageRank", *Proceedings of the 11th International Conference on World Wide Web*, pp. 517–526, 2002.

[HOL 75] HOLLAND J.H., *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, 1975.

[JOR 02] JORDAN M.I., WEISS Y., "Graphical models: probabilistic inference", in ARBIB M.A. (ed.), *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, 2002.

[KLE 02] KLEINBERG J., TARDOS E., "Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields", *Journal of the ACM (JACM)*, vol. 49, no. 5, pp. 616–639, 2002.

[LEE 92] LEECH G., "100 million words of English: The British National Corpus (BNC)", *Language Research*, vol. 28, no. 1, pp. 1–13, 1992.

[MAL 88] MALLERY J.C., Thinking about foreign policy: finding an appropriate role for artificially intelligent computers, Masters Thesis, MIT, 1988.

[MAN 14] MANION S.L., SAINUDIIN R., "An iterative sudoku style approach to subgraph-based word sense disambiguation", *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)*, pp. 40–50, 2014.

[MEN 14] MENAI M., "Word sense disambiguation using evolutionary algorithms – application to Arabic language", *Computers in Human Behavior*, vol. 41, pp. 92–103, 2014.

[MIH 04] MIHALCEA R., TARAU P., FIGA E., "PageRank on semantic networks, with application to word sense disambiguation", *Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics*, p. 1126, 2004.

[MOR 14] MORO A., RAGANATO A., NAVIGLI R., "Entity linking meets word sense disambiguation: a unified approach", *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 231–244, 2014.

[MOS 89] MOSCATO P., "On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms", *Caltech Concurrent Computation Program, C3P Report*, vol. 826, p. 1989, 1989.

[NAV 07a] NAVIGLI R., LAPATA M., "Graph connectivity measures for unsupervised word sense disambiguation", *International Joint Conference on Artificial Intelligence*, pp. 1683–1688, 2007.

[NAV 07b] NAVIGLI R., LITKOWSKI K.C., HARGRAVES O., "SemEval-2007 task 07: coarse-grained English all-words task", *Proceedings of the 4th International Workshop on Semantic Evaluations, Association for Computational Linguistics*, pp. 30–35, 2007.

[NAV 09] NAVIGLI R., "Word sense disambiguation: a survey", *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, p. 10, 2009.

[NAV 12a] NAVIGLI R., PONZETTO S., "BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network", *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[NAV 12b] NAVIGLI R., PONZETTO S., "Joining forces pays off: multilingual joint word sense disambiguation", *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1399–1410, 2012.

[PAG 99] PAGE L., BRIN S., MOTWANI R. *et al*., The PageRank citation ranking: bringing order to the web, Technical report, Stanford InfoLab, 1999.

[PAL 01] PALMER M., FELLBAUM C., COTTON S. *et al*., "English tasks: all-words and verb lexical sample", *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 21–24, 2001.

[PAN 02] PANTEL P., LIN D., "Discovering word senses from text", *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 613–619, 2002.

[PAT 06] PATWARDHAN S., PEDERSEN T., "Using WordNet-based context vectors to estimate the semantic relatedness of concepts", *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, vol. 1501, pp. 1–8, 2006.

[PRA 07] PRADHAN S.S., LOPER E., DLIGACH D. *et al*., "SemEval-2007 task 17: English lexical sample, SRL and all words", *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 87–92, 2007.

[REN 09] RENTOUMI V., GIANNAKOPOULOS G., KARKALETSIS V. *et al*., "Sentiment analysis of figurative language using a word sense disambiguation approach", *RANLP*, pp. 370–375, 2009.

[SAM 11] SAMMUT C., WEBB G.I., *Encyclopedia of Machine Learning*, Springer, Berlin, 2011.

[SIN 07] SINHAR S., MIHALCEA R., "Unsupervised graph-based word sense disambiguation using measures of word semantic similarity", *ICSC*, vol. 7, pp. 363–369, 2007.

[SMR 06] SMRŽ P., "Using WordNet for opinion mining", *Proceedings of the Third International WordNet Conference*, pp. 333–335, 2006.

[SNY 04] SNYDER B., PALMER M., "The English all-words task", *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 41–43, 2004.

[TAY 78] TAYLOR P.D., JONKER L.B., "Evolutionary stable strategies and game dynamics", *Mathematical Biosciences*, vol. 40, no. 1, pp. 145–156, 1978.

[TON 06] TONG H., FALOUTSOS C., PAN J., "Fast random walk with restart and its applications", *Proceedings of the Sixth International Conference on Data Mining*, pp. 613– 622, 2006.

[TRI 15a] TRIPODI R., PELILLO M., "WSD-games: a game-theoretic algorithm for unsupervised word sense disambiguation", *Proceedings of SemEval-2015*, pp. 329–334, 2015.

[TRI 15b] TRIPODI R., PELILLO M., DELMONTE R., "An evolutionary game theoretic approach to word sense disambiguation", *Proceedings of Natural Language Processing and Cognitive Science 2014*, pp. 39–48, 2015.

[TRI 17] TRIPODI R., MARCELLO P., "A Game-Theoretic Approach to Word Sense Disambiguation", *Computational Linguistics*, vol. 1, p. 43, 2017.

[VAP 98] VAPNIK V.N., *Statistical Learning Theory*, Wiley-Interscience, Hoboken, 1998.

[VIC 05] VICKREY D., BIEWALD L., TEYSSIER M. *et al*., "Word-sense disambiguation for machine translation", *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics*, pp. 771–778, 2005.

[VON 44] VON NEUMANN J., MORGENSTERN O., *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, 1944.

[WEA 55] WEAVER W., "Translation", in LOCKE W., BOOTH D. (eds), *Machine Translation of Languages*, vol. 14, Technology Press, MIT, Cambridge, 1955.

[ZHO 10] ZHONG Z., NG H.T., "It makes sense: a wide-coverage word sense disambiguation system for free text", *Proceedings of the ACL 2010 System Demonstrations, Association for Computational Linguistics*, pp. 78–83, 2010.

[ZHU 05] ZHU X., LAFFERTY J., ROSENFELD R., Semi-Supervised Learning with Graphs, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2005.

7

Use Your Mind and Learn to Write: The Problem of Producing Coherent Text

To produce written text can be a daunting task, presenting a challenge not only for high school students or second-language learners, but actually for most of us, including scientists and PhD students writing in their mother tongue. Text production involves several tasks: *ideation* (what to say?), *text structuring* (message grouping and linearization), *expression* (mapping of content onto linguistic forms) and *revision*. We will address here only one of them, *text structuring*, which is probably the most challenging task as it implies the grouping (chunking), ordering and linking of messages, which at the end of conceptual input lack this kind of information. Our goal is to find out whether part of this task can be automatized, the user providing a set of inputs (messages to be conveyed) and the computer then building automatically one or several *topic trees* from which the user will choose. While these trees still lack rhetorical information, functionally speaking they have a similar role as an outline: reduce the cognitive load of the writer and the reader. They help the writer to get some control over the information glut, telling him when to "say" what (order of sentences and paragraphs), and they help the reader to understand the functions of the different parts, i.e. how do the different parts relate to each other? As we can see, this is a very complex task. Having just begun to work on it, we will present here only preliminary results based on a very simple example and confined to a specific text type, *descriptions*. Yet, if ever this method works well for this type, it should also work, be it only partially, for other text types.

7.1. The problem

Spontaneous speech is a cyclic process involving a loosely ordered set of tasks: conceptual preparation, formulation, articulation [LEV 89, REI 00]. Given a goal, we have to decide *what to say* (conceptualization) and *how to say it* (formulation), making sure that the chosen elements, words, can be

Chapter written by Michael ZOCK and Debela Tesfaye GEMECHU.

integrated into a coherent whole (sentence frame) and do conform to the grammar rules of the language (syntax, morphology). During vocal delivery, in itself already quite a demanding task, the speaker may decide to initiate the next cycle, namely starting to plan the subsequent ideational fragment. In sum, speaking or acquiring this skill is a daunting task requiring the planning and execution of a number of subtasks. Given some goal, a speaker must plan *what to say* and *how to say* it, i.e. (a) *find* the right *words*; (b) determine an appropriate *sentence frame*; (c) put the chosen lemma in the right place; (d) add *function words*; (e) perform morphological adjustments; (f) articulate.

If speaking is difficult, writing is even a greater challenge, despite the huge amount of extra time. An author must not only know how to carry out most of the operations mentioned, but also be able to perform some additional operations which are not trivial at all. Some of them are at the *linguistic level* (cohesive devices: links, pronouns, choice of adequate determiner, etc.), and others are at the *conceptual level*: analysis and synthesis of knowledge[1], determination of information to provide and ensure reference (Henry Pu Yi, the last emperor, he), grouping, ordering and linking of messages, aggregation, i.e. merging syntactic constituents, etc. These last four operations are fundamental, as otherwise the reader may misunderstand or not understand at all. Being unable to see the connection between the parts, s/he cannot make sense of the whole. The document is perceived as a set of unrelated, i.e. incoherent segments. Yet, texts are characterized by the fact that goals, ideas and expressions are linked via a set of *rhetorical* (concession, rebuttal, etc.), *conceptual* (tense, cause–effect, set inclusion, etc.) and *linguistic* relations (anaphora, reference chains). Indeed, it is quite rare to see "texts" whose elements (propositions or sentences) are related only on the basis of statistical considerations (weight, frequency, etc.).

1 Reading the following sentence: "While there are many similarities between Japan and Germany there are also quite a few differences", it would be a mistake to believe that the expressed "facts" are stored like that in our memory. Indeed, what is expressed is probably the result of an analysis/synthesis of a large set of data concerning these two countries. Once we have performed this task, we may well conclude that despite the number of *commonalities* (discipline, work ethic, clean, well organized), there are also quite a few *differences* between the two countries: geographical location, religion, food (rice/potatoes, fish/meat), behavior (individualism vs. collective behavior), etc.

Text structuring is a particularly challenging task, because the ideas to be conveyed generally lack the links needed to build a coherent topic map, i.e. a tree showing which ideas go together and how the different chunks are related. Moreover, ideas tend to come to our mind in any order, i.e. via association [IYE 09]. Hence, in this case, "order of conceptual fragments" is a by-product of priming. It depends only on the relative associative strength between two items: a prime (doctor) and a probe (target, e.g. nurse). Obviously, this kind of order is very different from the one we see in ordinary texts, where the author guides the reader from some starting point (problem) to the conclusion (end point, solution). In conclusion, the order in which ideas come to our mind is fundamentally different from the order in which they will be conveyed in the final document, a well-structured text. Obviously, this transformation is not an easy task, yet, what makes things worse is the fact that the information needed to impose order on this data is generally absent in the conceptual input, i.e. the messages to be conveyed. This information needs to be inferred. This is probably the reason why writing is so much harder than speaking. Let us illustrate this via some concrete problems.

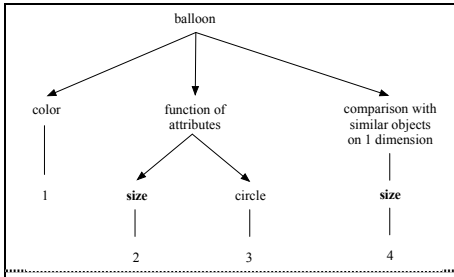
7.2. Suboptimal texts and some of the reasons

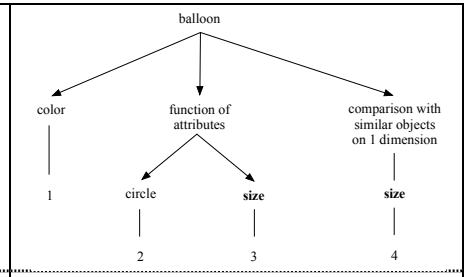
Texts are somehow like movies; they introduce and develop some objects over time. Having set the stage (context), the director introduces a topic, which s/he describes then from a chosen point of view, to move then on in various directions. To enable the onlooker to understand the movie (What is the point? How did we find the solution? What caused the coming about of some event?), the film director has to provide cues allowing the person watching the movie to grasp the topic, to see the details and to realize the evolution of the topic (topic changes, or return to the initial topic). In discourse, this is done via language, though the hardest parts are done in the brain. They are being taken care of by the reasoning and conceptual component: choice of the topic, relative order (development), type of connections, importance (focus) of the various elements at a given moment, etc. If sentences are like snapshots, texts are more like movies. They both have a framework, but the slides of a movie evolve over time. Hence, producing a good movie is, cognitively speaking, more demanding than taking a good snapshot.

While it is easy to tell whether a sentence is correct or not, it is not easy at all to do the same for a text. We may even wonder whether it makes sense to use the notion of correctness in this case. Texts span a wide spectrum, ranging from very well written to hardly understandable, with various stages in between. There are many reasons why a text may not read well: lack of coherence or cohesion, faulty reference, inadequate choice of a linguistic resource, etc., just a few examples to illustrate our point.

7.2.1. *Lack of coherence or cohesion*

Hearing someone say "John left for Taiwan to practice his French" may make us wonder about the connection between the two clauses, while the following, actually very similar sentence "John left for Taiwan to practice his Chinese" seems to be clear. We all understand that the second clause explains the reason of the first. The discourse is now coherent. Coherence is probably the single most important feature of text. Yet, even if texts lack coherence, they are hardly ever entirely incoherent. Readers sometimes even do not realize this, though it affects readability. Let us illustrate this via an example, where a cohesive document lacks *coherence*:





(1) The balloon was red and white striped. (2) Because this ballon was designed to carry men, it had to be large. (3) It had a silver circle at the top to reflect heat. (4) In fact, it was larger than any balloon John had ever seen. Text taken from [MCK 85]	(1) The balloon was red and white striped. (2) It had a silver circle at the top to reflect heat. (3) Because this ballon was designed to carry men, it had to be large. (4) In fact, it was larger than any balloon John had ever seen
Slightly incoherent paragraph	Fully coherent paragraph

Figure 7.1. *Two paragraphs varying in terms of coherence*

Both paragraphs are composed of the same messages, but in slightly different orders, affecting readability. In the text of the left-hand side, there is a conceptual disruption. The author starts by providing information concerning the "color" of the described object. Next, he mentions its "size" and "form", to make then again a comment concerning "size", a feature mentioned already earlier on. This caveat has been avoided in the text of the right-hand side which for these reasons reads better. It is more coherent.

Texts may be coherent, but lack *cohesion* which also makes them suboptimal. *Cohesion* is generally achieved via specific linguistic means: hypernyms, pronouns, rhetorical relations, etc. The text of Figure 7.2 here below was generated by a computer to answer the following question: "What is the location of Uruguay?" The text on the left-hand side is coherent as it clusters information concerning various features: (**a**) *position* (a-c: latitude and longitude); (**b**) *neighboring countries* (d-e: northeast and west) and (**c**) *natural borders* (f-h: rivers and coasts). Despite this fact, the text does not read well. It lacks integration. There are many repetitions, which could have been avoided by using cohesive devices (pronouns, links). This has been done in the text on the right-hand side, which for this reason is more fluent than its counterpart.

(a) The location of Uruguay is South America. (b) The latitude ranges from -30 to -35 degrees. (c) The longitude ranges from -53 to -58 degrees. (d) The northern and eastern bordering country is Brazil. (e) The western bordering country is Argentina. (f) The boundary is the Uruguay river. (g) The southeastern coast is the Atlantic ocean. (h) The southern coast is the Rio de la Plata. (Example taken from [CAR 70])	(1a-1c) Uruguay lies in South America between -30 to -35 degrees latitude and -53 to 58 degrees longitude. (2d-2e) Its neighbour countries are Brazil in the north and the east, and Argentina in the west. (3f) The boundary between the two of them is the Uruguay river. (4g-4h) Uruguay's natural borders are the Atlantic Ocean in the south-east and the Rio de la Plata in the south
Coherent but incohesive paragraph	Coherent and cohesive paragraph

Figure 7.2. *Two paragraphs varying in terms of cohesion*

7.2.2. *Faulty reference*

One of the first things children learn are referring expressions [MAT 07]: cat, dog, mouse. Yet, the production of these forms is not a simple task,

and there are various reasons for this. While the objects we talk about hardly ever change during our conversation, the linguistic means used for describing them do. In fact, they vary considerably, depending on whether we refer to an object for the first time or not: Once upon a time, there was a king named Henry. He had three daughters. Yet, the form depends not only on the referent's intrinsic features (gender: he/she), but also on competing elements. Imagine the scene here below, with the goal to refer to e_2.

Entity	*type*	*feature*$_1$	*feature*$_2$
e_1	cat	size: small	color: white
e_2	dog	size: big	color: black
e_3	mouse	size: small	color: gray

Table 7.1. *Description of the elements composing the scene*

In theory, we could use any of the following six forms: (1) a minimal description and basic-level term (dog), (2) a more specific word (Doberman), (3) a relational description (the dog *on* the lawn), (4) the referent's role (shepherd dog), (5) his proper noun (Fido) or (6) simply a pointer, i.e. a pronoun (it). While all these forms are correct, not all of them suit equally well the situation. The first example, using a basic level term (dog) and the definite article, fits best. Examples 5 and 6 (Fido, it) are underspecified and appropriate only under very specific circumstances: knowledge of the object's name, object currently in focus of attention, second mention, etc. Examples 2 and 3 are overspecified, as they provide more information than needed for identifying the intended object. For more details, see [ZOC 15b].

7.2.3. *Unmotivated topic shift*

Texts have both a hierarchical and linear structure. Entities (objects, topics) are introduced, and then developed. Since objects can be viewed from many perspectives, it is important to signal the viewpoint. The first entity is generally the perspective from which the described topic or scene is

viewed. Topics are generally confined to a paragraph, and unless being "broadcasted", they remain stable. Consider the following paragraph from one of O'Connor's novels [OCO 71]

> "Mrs. Shortley was watching a black car turn through the gate from the highway. Over by the toolshed, about fifteen feet away, the two Negroes, Astor and Sulk, had stopped work to watch. They **were hidden** by a mulberry tree but Mrs. Shortley knew they were there."

If in this last sentence the author had used the *active voice*, writing, "The mulberry tree **hid** them but...", the text would still be correct, though much less fluent. Indeed, the use of this particular grammatical option would introduce an *unmotivated topic shift*[2], changing the perspective from "Astor and Sulk" to the mulberry tree.

7.3. How to deal with the complexity of the task?

Language production can be an overwhelming task. To deal with its complexity, people have conceived a wide range of strategies and tools: decomposition of the whole task, incremental processing[3], use of external resources (encyclopedias, dictionaries) or material support (text editors, pen and paper), allowing them to jot down ideas, make an outline, write a draft, etc.

Writing is not only re-writing but also thinking. We cannot just dump our ideas onto the world; we must give them a certain structure and form. Scardamalia and Bereiter [SCA 87] introduced the important distinction between *knowledge-telling* and *knowledge-transforming*. Novices use this first strategy, expressing ideas basically in the order in which they come to their mind, making text production look like a linear process, while in reality it is a hierarchical one [MAN 87]. There is a main point, there are subsidiary points, transitions from one level to the next, etc. What the knowledge-

2 Note that topic shifts can be triggered by various linguistic devices: voice (active/passive) or verb choice (buy vs. sell). Proper control of topics is important, as the reader interprets the clause accordingly.

3 Psychologists [KEM 87] have studied *incremental processing* noting that speakers start expression (articulation) before having fully encoded, i.e. completely specified all the details of the message. What holds for speaking holds, of course, also for writing.

teller's strategy is crucially lacking is a purposeful reflection concerning the content, role and form of the elements to be used. The author seems to be a prisoner of his associations. Rather than stepping back and deciding from there "what" to say, "when" and "how", s/he jumps too quickly to verbal delivery. Moreover, the fact that this behavior is largely controlled by local associations (*what to say next?*) makes everything look as if it were on the same plane.

The second production mode, *knowledge-transforming,* is generally used by mature writers who develop far more elaborated networks (more connections, better integration among goals) than novices. *Knowledge-transforming* can be characterized by an inclusion in the writing process of reflective operations that transform intentional, structural and conceptual representations (gist). Analysis of thinking-aloud protocols [FLO 80] has shown that mature writers plan by globally working through a writing task at an abstract level before working through it at a more concrete level. During the text production process, problems are tackled both at the level of content (*what do I mean?*) and at the level of form (*how do I say it?*). Reflection on both levels during composition leads to the transformation of content and form, giving rise to new thoughts.

In sum, there is a fundamental difference between expert and novice writing with respect to the nature of planning. Planning by beginners is opportunistic and driven by local constraints, while the planning by experts is strategic: the writer plans much carefully *what* she wants to say and *how* to express "things". It is mainly the authors' goals that determine content, structure and form. As we can see, writing is not an easy task and we are wondering whether and to what extent computers could help. We will focus here only on one task, *coherence*, i.e. grouping messages by category. Yet before doing so, let us take a look at the work done by computational linguists, cognitive psychologists and rhetoricians on whose theories any good application rests.

7.4. Related work

Let us start by taking a look at the work done by computational linguists working on *text* generation [REI 00, BAT 16]. Their ambition consists of the

automatic production of texts based on messages and goals[4]. Since everyone seems to agree with the fact that texts are structured [MAN 87], this seems the right place to go. Alas, even there one will be disappointed. To avoid misunderstandings, the work produced by this community is important and impressive in many ways. Nevertheless, it seems to be based on assumptions incompatible with respect to our goal, which is to assist a writer in text production, i.e. help her or him to organize a set of ideas that prior to that point were a more or less random bunch of thoughts (at least for the reader).

Here are some of the reasons why we believe that this kind of work is not compatible with our goal. First of all, interactive generation (our case) is quite different from automatic text generation. Next, most text generators are based on assumptions that hardly apply in normal writing: (a) *all the messages* to be included in the final document are available at the very moment of building the text plan [HOV 91]; (b) ideas are retrieved *after* a text plan has been determined [MCK 85], or the two are done more or less in parallel [MOO 93]; (c) the links between ideas (messages) or the topics to be addressed are all known at the onset of building the text plan. This last point applies both to Marcu's work [MAR 97] and to data-based generators [REI 95].

Practically all these premises can be challenged, and none of them accounts for the psycholinguistic reality of composition, i.e. text production by human beings [DEB 84, BER 87, AND 96]. For example, authors often do not know the kind of links holding between ideas[5], neither do they always know the topical category of a given message[6]. Both have to be inferred. Authors have to discover the link(s) between messages and the nature of the topical category to which a message or a set of messages belongs. Both tasks are complex, requiring a lot of practice before leading to the skill of good writing (coherent and cohesive discourse).

4 This is often seen as a top-down process : goals triggering ideas, i.e. messages, which trigger words, which are inserted in some sentence frame, to be adjusted morphologically, etc.
5 The following two messages [(a) get married (x), (b) become pregnant (x)] could be considered as a *cause*, *consequence* or as a natural *sequence*.
6 What we mean by *topic* is the following. Suppose you were to write "foxes hide underground". In this case, a reader may conclude that you try to convey something concerning the foxes' "habits" (hide) or "habitat" (underground).

The above-mentioned work also fails to model the dynamic interaction between idea generation (messages) and text structure, [SIM 88] being arguably an exception. Indeed, a topic may trigger a set of ideas (top-down generation), just as ideas may evoke a certain topic (bottom-up), and of course, the two can be combined, a bottom-up strategy being followed by a top-down strategy (see Figure 7.3). This kind of interaction often occurs in spontaneous writing where ideas lead to the recognition of a topical category, which in turn leads to the generation of new data of the same kind. Hence, ideas or messages may have to be dropped. Not having enough conceptual material, the author may decide either not to mention a given fragment, to put it in a footnote, or to continue searching for additional material.

Another community interested in writing is that of psychologists. Clearly, a lot has been written on this subject[7]. Yet, despite the vast literature on composition and despite the recognition of the paramount role played by idea structuring (outlining) for yielding readable prose, little has been produced to clarify what it takes concretely speaking to achieve this goal (i.e. to help authors). Even the book series "Studies in Writing" [RIJ 96b][8] will tell you next to nothing concerning the topic we are interested in: how to find commonalities between conceptual fragments (ideas) to group them into chunks, or, how to "see" the hidden links between ideas.

In the remainder of this paper, we will present a small prototype trying to emulate the first strategy mentioned here above: to structure data, or discover potential structures in data (messages). Yet before doing so, we would like to spell out in more detail some of the assumptions underlying our work and show how they relate to what is known about the natural writing process.

7.5. Assumptions concerning the building of a tool assisting the writing process

As mentioned already, authors tend to use different strategies when writing: they start from topics or goals (top-down), from initially unrelated data or ideas (bottom-up), or they combine these two strategies. Bottom-up

7 Among others: [ALA 01, BER 87, FLO 80, KEL 99, LEV 13, MAT 87, OLI 01, RIJ 96a, RIJ 96b, TOR 99]. For more pointers, see: http://www.writingpro.eu/references.php

8 http://www.emeraldinsight.com/products/books/series.htm?id=1572-6304

activated ideas lead to the recognition of a subsumption category (Figure 7.3(b)), which in turn causes the activation of more data (top-down again, see Figure 7.3(c)).

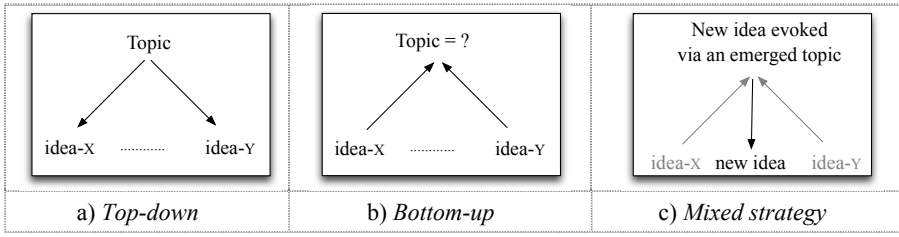


Figure 7.3. *Three strategies of discourse planning: top-down, bottom-up or both*

The first strategy is probably the most frequent one. Starting from a goal, authors seek relevant content (messages), organize it according to topical and rhetorical criteria, translate it into language and revise it. This is known as top-down planning. Note that revision can take place at any stage, and that, during the initial stage of conceptual planning (brainstorming), little filtering takes place. Authors are mainly keen on potentially interesting ideas. (Incidentally, this is why the term "brainstorming" better captures the reality of this situation than "idea planning".) It is only at the next step that contents are thoroughly examined. This may lead to a modification of the message base: some ideas will be dropped, others added. The result of this is generally a so-called outline or text plan, i.e. a definition of *what* is to be expressed *when*, i.e. in what *order*.

Another strategy involves going the opposite way. Starting from the ideas coming spontaneously to the authors' mind (bottom-up planning), s/he will try to group them into sets (topical trees) and to link these clusters. In this kind of bottom-up planning, the structure or topic emerges from the data. These topics may act as seeds, eventually triggering additional material (mixed strategy). Bottom-up planning is a very difficult problem (even for people). Yet, this is the one we are interested in. A question remains on the basis of what knowledge writers know which ideas cohere, i.e. what goes with what and in what specific way? Suppose you have an assignment asking you to write a small document about "foxes" and their similarities and differences compared with "wolves" or "coyotes", two animals with which they are sometimes confused. This might trigger search for information concerning "foxes", possibly yielding a set of messages like the

one shown in Figure 7.3(a). For the time being, it does not really matter where these ideas come from (author's brain, external resources or others), what we are interested in here is to find an answer to the following questions: (a) How does the author group these messages or ideas into topical categories? (b) How does s/he order them within each group? (c) How does s/he link and name these chunks? (d) How does s/he discover and name the relations between each sentence or chunk?

We will focus here only on the first question (topical clustering), assuming that (a) messages will be grouped if they have something in common, and (b) messages or message elements do indeed have something in common. The question is how to show this. Actually, this can be either hard or fairly trivial, as in the case of term identity. Imagine the following inputs: (a) give (I, dog$_1$, my_son) and (b) like_to_chase (dog$_1$, milk-man). Since these two propositions share an argument (dog$_1$), they can be clustered, yielding two independent clauses (I've given my son a dog. He likes to chase the milk-man.), or a subordinate, i.e. relative clause (I've given my son a dog who likes to chase the milkman). Of course, you could also topicalize the "dog", producing the following sentence: "The dog I've given to my son likes to chase the milkman". Which of these forms is the most adequate one depends, among other things, on the context (surrounding sentences) and the discourse goal: what do you want to stress or highlight?

The question of how to reveal commonalities or links between data in the non-obvious cases remains. We can think of several methods. For example, we could try to enrich input elements by adding information (features, attribute-values, etc.) coming from external knowledge sources: corpora (co-occurrence data, word associations), dictionaries (definitions), etc. Another method could consist of determining similarities between message elements (words). This is the one we have used, and we will explain it in more depth here below (section 7.6). Once such a method has been applied, we should be able to cluster messages by category, even though we may not be able to give it a name. The name may be implicit, and name-giving may require other methods.

The result of this will be one or several topic trees, grouping (ideally) all inputs. While different trees may achieve different rhetorical goals (the focus being different), all of them ensure coherent discourse. The effect of these variances can probably only be judged by a human user, who shall pick the one best fitting his or her needs. While our developed software will not be

able to achieve this goal, i.e. build a structure that conceptually and rhetorically matches the authors' goals, it should nevertheless be able to help the user perceive conceptual coherence, hence allow him to create a structure (topic tree) where all messages cohere, something that not all grown-up human beings are able to do. Concerning goals and bottom-up planning, consider the following.

Goals can be of various sorts. They can be coarse grained ("convince your father to lend you his car") or more fine-grained, relating to a specific topic: describe an animal and show how it differs from another one with which it is often confused (alligator-crocodile; fox-coyote/wolf). *Messages* may feed back on the conceptual component, altering messages or goals (addition, deletion, modification). This cyclic process between top-down and bottom-up processing is a very frequent case in human writing. We will focus here only on the latter, confining ourselves to propositions composed of two place predicates. These will be the inputs for which we try to check whether there is a commonality or link between them. Of course, even the linking of simple propositions may be a very complex problem. Think of causal relations that can be viewed as a systematic correlation between two events, or a state and an event[9]. Since these cases require a special approach, we will not deal with them here.

7.6. Methodology

We present in this section a description of the method used to allow for the kind of grouping mentioned here above. Messages can be organized on various dimensions and according to various viewpoints: *conceptual* relations (taxonomic, i.e. set inclusion, causal, temporal, etc.), *rhetorical* relations (concession, disagreement), etc. We will focus here only on the former, assuming that messages can at least to some extent be organized via the semantics[10] of their respective constituent elements.

9 The perception of the causal relationship between the underlined elements in – "Be careful, the road may be *dangerous*. They've just announced a *Typhoon*." – supposes that we know that Typhoons are dangerous.

10 Of course, the term semantics can mean many things (association, shared elements between a set of words, etc.), and which of them an author is referring to needs to be made explicit.

Put differently, in order to reveal the relative proximity or relation between a set of messages, we may consider the similarity of some of their constituent elements. Summing similarity values is a typical component of a vector space model and has been well described by [WID 04, MAN 08]. Concerning "similarity", we need to be careful though, as the words' similarity does not guarantee "relatedness"; it may even be one of its preconditions. Indeed, many researchers have used this feature for sentence similarity detection [BUL 07, TUR 06], but most of them based their analysis on the surface form, which may lead to erroneous results, because similar meanings can be expressed via very different syntactic categories (e.g. "use for" vs. "instrument", "have" vs. "her"). Likewise, a given form or linguistic resource, say the possessive adjective, may encode very different meanings. Compare – his car versus his father versus his toe – which express quite different relations: ownership, family relationship, inalienable part of the human body.

What we present here is a very preliminary work. Hence, our method is designed to address only very simple cases, two-place predicates, i.e. sentences composed of two nouns (a subject and an object) and a (linking) predicate. Given a set of these kind of inputs, our program determines their proximity regardless of their surface forms. The sentences will be clustered on the basis of semantic similarity between the constituent words. This yields a tree whose nodes are categories (whose type should ideally be expressed explicitly, e.g. food, color, etc.) and whose leaves are the messages or propositions given as input.

In the following sections, we will explain in more detail our approach by taking the inputs shown in Figure 7.4(a) to illustrate our purpose. The goal is to cluster these messages by topic to create a kind of outline or topical tree. Indeed, {1, 4} address *physical features* (appearance), {2, 6} provide spatial information, the place where foxes live or hide (*habitat*), while {3, 5, 7} deal with their *habits*. This last category can be split into two subtopics, in our case, "theft" {3} and "consumption" {5, 7}. The result of this analysis can be displayed in the form of a tree (Figure 7.4(b))[11].

11 Note that generally we can come up with more than one tree. Any set of data allowing for multiple analyses (depending on the point of view), and multiple rhetorical effects.

1° resemble (foxes, dog)
2° live_in (foxes, woods)
3° steal (foxes, chicken)
4° be (foxes, red)
5° eat (foxes, fruits)
6° hide (foxes, underground)
7° eat (foxes, eggs)

Figure 7.4(a) *Conceptual input (messages)*

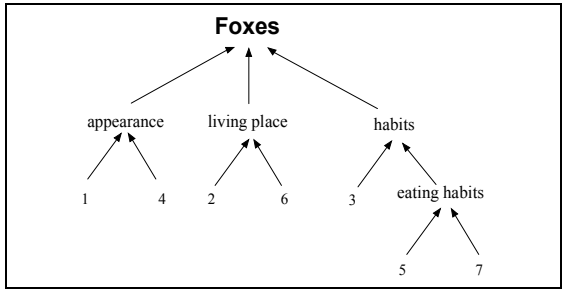


Figure 7.4(b) *Clustered output (topic tree)*

In order to achieve this result, we have defined an algorithm carrying out the steps referred to in Table 7.2. We will describe and explain them in more depth in the following sections. Note that what we called messages here above is now called sentence which is processed by a parser.

1) Determine the role of words, i.e. perform a syntactic analysis;
2) Find potential seed words;
3) Align words playing the same role in different sentences;
4) Determine the semantic proximity between the aligned words;
5) Determine the similarity between sentences;
6) Group sentences according to their semantic affinity (similarity).

Table 7.2. *Main steps for topic clustering*

7.6.1. *Identification of the syntactic structure*

The goal of this step is to identify the dependency structure of the sentence. This information will be used later on (**a**) to identify the semantic

seeds (see section 7.6.2), (**b**) to align words playing a similar role and (**c**) to identify the role of the different elements of the underlying proposition, i.e. the respective predicate, subject or object. To obtain this information, we used the Stanford parser[12]. For example, the input "Foxes eat fruits" would yield the following output:

Tagging: **Foxes**/NNS **eat**/VBP **fruits**/NNS./.

Dependencies: N_{subj} (eat, foxes); D_{obj} (eat, fruits)

Of all these outputs, we are concerned here only with N_{subj} and D_{obj} in order to determine the main elements of the message: the subject, object and the main verb, or, in propositional terms [predicate (argument$_1$, argument$_2$)]. Next, we used the similarity of the parts (words) to determine the similarity of the wholes (sentences).

7.6.2. *Identification of the semantic seed words*

As mentioned already, in order to reveal the proximity or potential relation between two or more sentences, we can try to identify the similarity between the respective constituent words. We need to be careful though. If we do this by taking into account only the similarity values of the respective (pair of) words, we may bias the analysis and get incorrect results.

There are several problems at stake. For example, the number of identical words does not necessarily imply relatedness or similarity. Actually, two sentences may be composed of exactly the same words, and still mean quite different things, compare: "Women without their men are helpless" versus "Men without their women are helpless". Given the fact that such cases are quite frequent in natural language, we decided not to rely on (all) the words occurring in a sentence, or to use a "bag of words" approach (sentence without stop words). We preferred to rely only on specific words, called seeds, to compare the similarity of different sentences. We consider seed words to be elements conveying the core meaning of a sentence. For example, for the two sentences here above we could get the following seeds: (a) without (man, women); (b) without (women, men), which reveal quite readily their difference.

Our idea of choosing seed words seems all the more justified as different kinds of words (lexical categories) have different statuses: some words

12 http://nlp.stanford.edu/software/lex-parser.shtml

conveying more vital information than others. Nouns and verbs are generally more important than adjectives and adverbs, and each one of them normally conveys more vital information than any of the other parts of speech[13]. We assumed here that the core information of our sentences is presented via the nouns (playing different roles: subject, object) and the verb linking them (predicate). We further assumed that dependency information was necessary in order to be able to carry out the next steps. In the following two sentences, (a) "Foxes hide underground" and (b) "Foxes hide *their prey* underground", a "bag of word" method or a simple surface analysis would not do, as neither of them reveals the fact that the object of hiding ("fox" vs. "prey") is different in each sentence, a fact that needs to be made explicit.

To avoid this problem, we used the dependency information produced by the parser, which allowed us to determine the role of the nouns (deep-*subject, deep-object*) and the predicate (verb) linking the two. For example, this reveals the fact that the following two sentences are somehow connected: "*Foxes eat eggs*" and "*Foxes eat fruits*". In both cases, the concept "fox" is connected to some object ("egg" vs. "fruit") via some predicate, the verb "*eat*". The core of these two sentences is identical. Both of them tell us something about the foxes' diet or eating habits (egg, fruits). Note that while this method does not reveal the nature of the link (diet, food), it does suggest that there is some kind of link (both sentences talk about the very same topic: food). Hence, syntactic information (part of speech, dependency structure) is precious as it allows us to identify potential seed words that will be useful for subsequent operations.

7.6.3. *Word alignment*

In order to compare sentences in terms of similarity, we not only need a method for doing so, but also need the data to be in a comparable form. Hence, we need the input to be given in a standardized form or we need to carry out some normalization. The latter can be accomplished to some extent via a dependency parser that reveals the roles played by different words. We can now align the words of the various sentences and compare those playing the same semantic role.

Word alignment consists not simply in finding identical words in different sentences, but rather in finding and aligning words playing the

13 Note that we do not consider "connectors" (yet, despite, because) here, as they are not known at this stage.

same role in these sentences. This means in our case that we have to compare, say, the subject of one sentence with the subject of another, and do the same for the other syntactic categories or semantic roles (verbs, deep-objects, etc.). To allow for this, we rely again on the dependency information produced by the parser (section 7.6.1). Note that our example showed only surface relations (subject, object, etc.), while ideally we need information in terms of deep-case roles: agent, beneficiary, etc. [FIL 68]. Applied to our examples, "Foxes eat fruit" and "Foxes eat eggs", it is clear that "fruit" can be aligned with "eggs", since both nouns play the same role.

Note that we also need to be able to detect synonyms or semantic equivalences: "instrument ≡ is used for"; "resemble ≡ be alike", "for example ≡ somehow ≡ like", etc. These words are very useful and could be used as topic-signatures [LIN 00], hence seed words. Note that such information is obtained indirectly in our approach via the *vector space model* which is briefly described in the next section.

7.6.4. *Determination of the similarity values of the aligned words*

While there are various ways to detect links between sentences or words (e.g. shared features or associations), two obvious ones are coreference and class-membership. See our example in Figure 7.4(a), where the two sentences – ("Foxes eat eggs"; "Foxes eat fruits") – have an identical referent, the actor "fox", and two different instances of the same class, the generic element "food".

As mentioned already, in order to compare sentences in terms of their meaning, the compared structures must have a common format. Similarity of meaning supposes, of course, that we are able to extract somehow the meaning of the analyzed objects (sentences, words). Yet, word meanings depend on the context in which a word occurs. Words occurring in similar contexts tend to have similar meanings. This idea known as the "distributional hypothesis" has been proposed by several scholars, e.g. [FIR 57, HAR 54, WIT 22]. For surveys, see [SAH 08, DAG 99], or (http://en.wikipedia.org/wiki/Distributional_semantics).

Since we try to capture meaning via word similarity, the question of how to operationalize this notion arises. One way of doing so is to create a vector space composed of the target word and its neighbors [LUN 96]. The meaning of a word is represented as a vector based on the co-occurring

words. In the following two sentences – "Foxes eat eggs" and "Foxes eat fruits" – we have four distinct tokens or words: foxes, eat, fruits and eggs. Hence, we constructed a vector for each one of them by considering their co-occurrences in COHA (Corpus of Historical American English), a part of speech tagged n-gram corpus of 400 million words [MAR 11]. This allowed us to apply the vector space model in order to compute the degree of similarity between a set of words. To this end, we computed the distance (cosine) of the respective vectors. Let us suppose that there are only four words co-occurring with "fruit" and "egg" ("juice, vitamin, price, eat" and "chicken, protein, eat and oval"), then the vector for fruit would be "juice, vitamin, price, eat" and the vector for egg would be chicken, protein, eat, oval.

Note that we will also count the frequency of the co-occurrence. To compute the similarity between two vectors, we computed the cosine of their angle. Hence, we constructed such vectors for all major words of our sentences. For the example here above, we have four vectors, one for each of the words occurring in both sentences: fox, eat, fruit and eggs. Note that the words in the vectors are replaced by their weight, i.e. a numerical value representing the meaning of the respective concepts. For instance, for the *fruit* vector, all of the following words, "juice, vitamin, price, eat" are replaced by a numerical value (weight). For details, see step 1 here below.

Consider the following example that measures the proximity between *fruit* and *eggs* by using the co-occurrence information gleaned from COHA, yielding the matrix shown here below.

I	word space	$\vec{T}_{1\,\text{fruit}}$	$\vec{T}_{2\,\text{eggs}}$
1	bird	0	1
2	hummus	0	1
3	food	1	1
4	incubator	0	1
5	banana	0	1
6	store	1	0
7	gene	1	1

Table 7.3. *Sample word space matrix*

The vectors are built by carrying out the following four steps:

1) Extract words co-occurring with *fruit* in a predefined window. In our experiments, we considered the window to be six content-bearing words of the sentence containing the target, hence, three terms preceding and following the term *fruit* that would yield in this case the following list of words: banana, food, gene and store.

2) Do the same for *eggs*, that would yield: bird, hummus, food, incubator and gene.

3) Build the corresponding vectors for each word (*fruit, egg*) based on its co-occurrences (see the table here above). In its simplest form, the vectors for each word are built on the basis of their co-occurrences. A term (*fruit, eggs*) receives frequency values depending on the number of times it co-occurs with another term within the defined window, which would yield the following vectors for *fruit* [0,0,1,0,0,1,1] and *vehicle* [1,1,1,1,1,0,1]. Note that the words here are referred to via their index, e.g. their position in the word space. Hence, the first position refers to the first term, the second to the second, and the last to the last one in the matrix. Unlike in the example given here above, we use in our experiments the weighted frequency of the words' co-occurrences rather than binary values.

4) Measure the distance between the vectors. In order to quantify the similarity between two words, we have to compute the cosine similarity of their respective vector representations. This is done in the following way:

Take a pair of words, say *fruit* (T_1) and *eggs* (T_2), with their respective vectors and weights, and carry out the steps described here below. To determine the similarity between "fruit" and "eggs" we compute their cosine similarity, i.e. the angle between their vectors. The cosine similarity value is computed in the following way:

1) sum the product of the weight of the terms in the respective vectors ($\vec{T}_1$ and $\vec{T}_2$);

2) sum the square of the weights of each vector term of $\vec{T}_1$ and $\vec{T}_2$;

3) take the square root of the product of the results of step 2;

4) divide the result of step 1 by the result of step 3.

We have used this method already for another task, the automatic extraction of part–whole relations [ZOC 15a]. Since then, we have extended it to allow for the computation of similarity between words. The method consists basically of two operations: creating a vector for all words and identifying the similarity of the aligned words.

Step 1: *Creation of a vector for all words based on co-occurrence information*

The co-occurrence information is gleaned from COHA. The vector is built on the basis of a word's co-occurrence within a defined window (phrase, sentence and paragraph). Since different words make different contributions, we assign weights to reflect their relative contribution in terms of relevance allowing determination of the meaning of a sentence or word. Hence, meanings are expressed via a weight, which may depend on the context of a given term, a factor that needs to be taken into account. The formula used to assign the weight is very similar to the TF-IDF. Since our objective is determining the similarity between words, the document–term matrix in the conventional TF-IDF is adopted to term–term matrix in our case. In order to determine the weight (`W`), we use the percentage of the term's **c**o-occurrence **f**requency (`TCF`) with respect to a given concept out of the **tot**al **n**um**b**er of **c**o-occurrences (`TOTNBC`) of the term with any other term in the corpus. For example, in order to determine the weight of the term *egg* (one of the words in the "chicken vector") for the vector of "chicken", we count the co-occurrence frequency of *chicken* with *egg* and then divide this result by the total number of co-occurrences of the term *egg* with any other term in the corpus. We did the same for all relevant terms. The above-described operations can be captured via the following formula, which is used to determine the weight (W) of a given word for determining the meaning of another co-occurring term:

`W = TCF-yz/TOTNBC-xy,` where

`TCF-y` is the frequency, i.e. the number of times `y` co-occurs with `z;`

`TF-y` is the total frequency y in the corpus.

To build a vector for a given concept, we use the weighted value of all co-occurring words. Hence, we calculated the relative weight of each word of the vectors in defining the meaning of the term for which the vector was built.

Step 2: *Identification of the cosine similarity between vectors of the aligned words*

The similarities between words are computed on the basis of the cosine of the words' vectors. Note that the similarity value is calculated only for the aligned words.

7.6.5. *Determination of the similarity between sentences*

The meaning of a sentence can (at least to some extent) be obtained via the combined meanings of the constituent elements, words. Having identified in section 7.6.4 the similarity values between the aligned words, we now build a matrix showing their respective similarity values. The rows and columns of the vectors are built on the basis of co-occurring words and the cells contain their similarity values. In order to identify the similarity between two sentences, we add the similarity values of the component words and then compute the average to derive a single similarity value between the sentences. Hence, in order to identify the degree of similarity between a pair of sentences, we sum up the respective similarity values of their subjects, verbs and objects, and then divide the result by 3 in order to get the average.

	resemble	eat	are	live	steal	hide
resemble						
Eat	0.293					
Are	0.550	0.152				
Live	0.210	0.365	0.139			
Steal	0.428	0.392	0.210	0.306		
Hide	0.527	0.430	0.240	0.631	0.450	

Table 7.4. *Sample word similarity matrix of co-occurrences*

With respect to our fox example (Figure 7.4), all messages apart from the third one are clustered this way. Message 3 is clustered with 5 and 7 according to the algorithm presented in the next section.

7.6.6. *Sentence clustering based on their similarity values*

As mentioned already, our strategy consists of the creation of a tree based on the similarity values of the sentences given as input. Sentences are clustered in three steps on the basis of the similarity value of the subjects, the verb and the objects. Accordingly, sentences talking about different topics, say "foxes" and "fruits", are placed in different clusters. At the next cycle, each group is further clustered depending on the topic (habit, physical appearance, etc.), which may be signaled via the verb or the object. The clustering algorithm is given below.

1) Take any sentence of the considered pool of inputs and search for another one whose topic similarity (i.e. value of the subject) is closer to the target than any of its competitors to form a cluster. Similarity values are obtained via the word-similarity-matrix (see section 7.6.5 and table 7.4).
2) Continue to cluster the sentences of the groups obtained so far by using the similarity values of the verb linking the subject and the object.
3) Perform the same operation as in step 2 based on the similarity of the objects.
4) Repeat steps 1 to 3 until all the sentences, or the greatest possible number of sentences are clustered.
5) Create a link between the clusters based on the respective similarity values of the verb and the object.

Table 7.5. *The clustering algorithm*

7.7. Experiment and evaluation

In order to test our system, we used a text collection of 28 sentences. The test set contains four groups of sentences talking respectively about "foxes", "fruits", and "cars". The last set, called rag bag, is composed of topically unrelated sentences. It is used only for control purposes.

Topic$_1$ *Fox*	Topic$_2$ *Fruits*
1) Foxes resemble dogs.	8) An apple a day keeps the doctor away.
2) Foxes live in the woods.	9) Apples are expensive this year.
3) Foxes steal chicken.	10) Oranges are rich in vitamin C.
4) Foxes are red.	11) The kiwi fruit has a soft texture.
5) Foxes eat fruits.	12) Grapes can be eaten raw.
6) Foxes hide underground.	13) Grapes can be used for making wine.
7) Foxes eat eggs.	14) The strawberries are delicious.
Topic$_3$ *Cars*	Topic$_4$ *ragbag*
15) A car is a wheeled motor vehicle.	22) Olive oil is a fat obtained from olives.
16) Cars are used for transporting passengers.	23) Playboys usually have a lot of money.
17) Cars are mainly designed to carry people.	24) A finger is a limb of the human body.
18) The first racing cars amazed the automobile world.	25) Apple is the name of a software company.
19) Cars typically have four wheels.	26) Eau Sauvage is a famous perfume.
20) Cars are normally designed to run on roads.	27) Wine is an alcoholic beverage made from fermented grapes or other fruits.
21) Cars also carry their own engine.	28) IBM is an American corporation manufacturing computer hard ware.

Table 7.6. *Our test sentences*

The system's task is now to integrate as many messages as possible. This will yield a tree containing as many branches as there are topics, in our case four. At the next cycle, the system will try to create subcategories, i.e. branches are further divided, or, viewed in the opposite direction, messages are clustered in more specific categories (habits, living places, etc., in the fox group). Whether this is feasible depends, of course, on the message elements. Since the function of the control group (the set of topically unrelated sentences) is only to check the system's accuracy, none of its sentences should appear in any other group than the "control group".

Once this clustering is done, we can determine the system's performance by counting the number of sentences assigned properly in the tree. For the

evaluation, we used the classical metrics, defining *precision*, *recall* and *F-measure* in the following way:

recall (Number of sentences correctly assigned to the valid cluster/Total number of sentences);

precision (Number of sentences correctly assigned to the valid cluster/ Number of sentences clustered);

f-measure (2/ [(1/precision) + (1/recall)]).

We obtained the following results: 22 out of the 28 sentences are placed in the correct cluster, six occur in the wrong place. For example, the sentences 25 and 27 are both placed in the fruit cluster (topic 2), while they should not. This is due to the fact that our method does not take into account the semantics of the string "apple", which can refer both to the fruit and to the computer manufacturing company located at Cupertino. The same holds for "wine", which can be both an alcoholic drink and a fruit.

We have also evaluated our system by further clustering the sentences within each topic based on their similarity. All the sentences of topic 1 are clustered correctly, with sentences 1-4, 2-6 and 5-7 forming three clusters. Sentence 3 is more closely related to the clusters containing 5 and 7 than to any other cluster. All the sentences in group 2 are clustered and two of them (10 and 14) are grouped in a more specific category. However, we do have some problems as some items appear in the wrong place. Sentences 25 and 27 are intruders, dealing with a different topic. Hence, they should not be in this category. The same holds for sentence 9, which is put in the (sub)group of sentences 10 and 14. This is clearly wrong, since it is about the *computer company* and not about "apple", the *fruit*. On the other hand, sentence 11 should be there, i.e. in the same cluster as 10 and 14. Yet it is not. It is placed elsewhere, standing like a loner, which, of course, is a mistake. In topic 3 (cars), all the sentences are clustered correctly, and two sentences (15, 16) are placed in a more specific category. However, the system failed to cluster 19 and 21 together.

Given the above described results, the system has a recall, precision and f-measure of 78.6%, if we consider only the sentences placed in the correct position in the tree. It is interesting to note that some of the sentences placed in the wrong category have a similarity value fairly close to the one of the

correct cluster. Actually, most of them have the second highest similarity value. Note also that, at this point, the clusters, i.e. the nodes of the tree, are not labeled. Whether this could be achieved via topic signatures [LIN 00] remains to be shown and is clearly a work for the future.

7.8. Outlook and conclusion

Organizing thoughts and expressions in such a way that discourse flows is a challenge for everyone. The difficulties are of various sorts. We have to acquire knowledge concerning linguistic resources (lexicon, grammar, textual devices: anaphora, etc.) and learn how our mind interprets the use of each one of them or the combination of conceptual entities (propositions). Once we have acquired this, effects can be used as goals, the normal starting point in communication.

A lot of work has been done on language production, yet, most of it from an engineering perspective [REI 00, BAT 16], i.e. fully automated text generation. Unfortunately, next to nothing has been done to assist the writer (our goal), though there is clearly a need for it. Note that the conditions under which an author works are far more multifarious and open-ended than the ones computers have to struggle with [AND 96]. Hence, instead of presenting a fully automatic solution as most computational linguists do, we propose here an interactive one. More precisely, we try to build an authoring tool where the computer helps the writer to organize his/her thoughts. Given a set of messages (user's knowledge state) and possibly a goal, the system structures the data in such a way that, conceptually speaking, the output is coherent, yielding one or several conceptual trees. Since this is quite a complex task, we have started with a very simple set of data. Nevertheless, we have tried to go beyond the obvious (co-references), to reveal some of the hidden and distributed information.

Our next steps will involve dealing with other conceptual relations and with the labeling of topical categories. We would also like to explore the idea of the information associated with the seed words. This should allow us to see on what basis (origin of the information on which) topical clustering is performed. As data can be grouped in many ways (depending on the chosen criteria or view points), and as various orders are likely to yield different effects, it would be nice to show the relationship between ordering and rhetorical effects.

Concerning the method, we could have also considered the following strategy: take a set of well-written texts of a specific type (description), extract its sentences, normalize and scramble them and have the computer try to reorganize them to produce a coherent whole, matching, as well as possible, the initial document (gold standard). Yet, having thought about this strategy too late (see, e.g., [BAR 08, BOL 10, LAP 06, WAN 06]), we used a different approach.

7.9. Bibliography

[ALA 01] ALAMARGOT D., CHANQUOY L., *Through the Models of Writing*, Kluwer, Dordrecht, 2001.

[AND 96] ANDRIESSEN J., DE SMEDT K., ZOCK M., "Discourse planning: empirical research and computer models", in DIJKSTRA T., DE SMEDT K. (eds). *Computational Psycholinguistics: AI and Connectionist Models of Human Language Processing*, Taylor Francis, London, 1996.

[BAR 08] BARZILAY R., LAPATA M., "Modeling local coherence: an entity-based approach", *Computational Linguistics*, vol. 34, no. 1, pp. 1–34, 2008.

[BAT 16] BATEMAN J., ZOCK M., "Natural language generation", in MITKOV R. (ed.), *Handbook of Computational Linguistics*, Oxford University Press, London, 2016.

[BER 87] BEREITER C., SCARDAMALIA M., *The Psychology of Written Composition*, Erlbaum, Hillsdale, 1987.

[BOL 10] BOLLEGALA D., OKAZAKI N., ISHIZUKA M., "A bottom-up approach to sentence ordering for multi-document summarization", *Information Processing and Management*, vol. 46, no. 1, pp. 89–109, 2010.

[BUL 07] BULLINARIA J.A., LEVY J., "Extracting semantic representations from word co-occurrence statistics: a computational study", *Behavior Research Methods*, vol. 39, pp. 510–526. 2007.

[CAR 70] CARBONELL J.R., Mixed-initiative man-computer instructional dialogues, PhD Thesis, Massachusetts Institute of Technology, 1970.

[DAG 99] DAGAN I., LEE L., PEREIRA F., "Similarity-based models of co-occurrence probabilities", *Machine Learning*, vol. 34, no. 1–3 special issue on Natural Language Learning, pp. 43–69, 1999.

[DEB 84] DE BEAUGRANDE R., *Text Production: Towards a Science of Composition*, Ablex, Norwood, 1984.

[DEM 08] DE MARNEFFE M.C., MANNING C.D., Stanford typed dependencies manual, Technical report, Stanford University, 2008.

[FIL 68] FILLMORE C., "The case for case", in BACH E., HARMS R. (eds), *Universals in Linguistic Theory*, Holt, Rinehart and Winston, New York, 1968.

[FIR 57] FIRTH J.R., "A synopsis of linguistic theory 1930–1955", in *Studies in Linguistic Analysis*, Philological Society, Oxford, 1957.

[FLO 80] FLOWER L., HAYES J.R., "The dynamics of composing: making plans and juggling constraints", in GREGG L., STEINBERG E.R. (eds), *Cognitive Processes in Writing*, Erlbaum, Hillsdale, 1980.

[HAR 54] HARRIS Z., "Distributional structure", *Word*, vol. 10, no. 23, pp. 46–162, 1954.

[HOV 91] HOVY E.H., "Approaches to the planning of coherent text", in PARIS C.L., SWARTOUT W.R., MANN W.C. (eds), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Kluwer Academic, 1991.

[IYE 09] IYER L.R., DOBOLI S., MINAI A.A. *et al*., "Neural dynamics of idea generation and the effects of priming", *Neural Networks*, vol. 22, no. 5–6, pp. 674–686, 2009.

[KEL 99] KELLOGG R., *Psychology of Writing*, Oxford University Press, New York, 1999.

[KEM 87] KEMPEN G., HOENKAMP E., "An incremental procedural grammar for sentence formulation", *Cognitive Science*, vol. 11, pp. 201–258. 1987.

[LAP 06] LAPATA M., "Automatic evaluation of information ordering", *Computational Linguistics*, vol. 32, no. 4, 2006.

[LEV 89] LEVELT W., *Speaking: From Intention to Articulation*, MIT Press, Cambridge, 1989.

[LEV 13] LEVY C.M., RANSDELL S., *The Science of Writing. Theories, Methods, Individual Differences and Applications*, Routledge, Abingdon, 2013.

[LIN 00] LIN C.-Y., HOVY E., "The Automated Acquisition of Topic Signatures for Text Summarization", *Proceedings of the COLING Conference*, pp. 495–501, 2000.

[LUN 96] LUND K., BURGESS C., "Producing high-dimensional semantic spaces from lexical co-occurrence", *Behavior Research Methods, Instruments, and Computers*, vol. 28, no. 2, pp. 203–208, 1996.

[MAN 87] MANN W.C., THOMPSON S.A., "Rhetorical structure theory: a theory of text organization", in POLANYI L. (ed.), *The Structure of Discourse*, Ablex, Norwood, 1987.

[MAN 08] MANNING C.D., RAGHAVAN P., SCHÜTZE H., *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, 2008.

[MAR 97] MARCU D., "From local to global coherence: a bottom-up approach to text planning", *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 629–635, 1997.

[MAR 11] MARK D., "N-grams and word frequency data", Corpus of Historical American English (COHA), 2011.

[MAT 87] MATSUHASHI A., "Revising the plan and altering the text", in MATSUHASHI A. (ed.), *Writing in Real Time*, Ablex, Norwood, 1987.

[MAT 07] MATTHEWS D.E., LIEVEN E., TOMASELLO M., "How toddlers and preschoolers learn to uniquely identify referents for others: a training study", *Child Development*, vol. 78, no. 6, pp. 1744–1759, 2007.

[MCK 85] MCKEOWN K.R., *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge University Press, Cambridge, 1985.

[MOO 93] MOORE J.D., PARIS C.L., "Planning text for advisory dialogues: capturing intentional and rhetorical information", *Computational Linguistics*, vol. 19, no. 4, 1993.

[OCO 71] O'CONNOR F., "The displaced person", in *The Complete Stories*, Macmillan, London, 1971.

[OLI 01] OLIVE T., LEVY C.M., *Contemporary Tools and Techniques for Studying Writing*, Kluwer, Dordrecht, 2001.

[REI 95] REICHENBERGER K., RONDHUIS K.J., KLEINZ J. *et al.*, "Communicative goal-driven NL generation and data-driven graphics generation: an architectural synthesis for multimedia page generation", *9th International Workshop on Natural Language Generation*, Niagara on the Lake, 1995.

[REI 00] REITER E., DALE R., *Building Natural Language Generation Systems*, Cambridge University Press, Cambridge, 2000.

[RIJ 96a] RIJLAARSDAM G., VAN DEN BERGH H., "The dynamics of composing – an agenda for research into an interactive compensatory model of writing: many questions, some answers", in LEVY C.M., RANSDELL S. (eds), *The Science of Writing: Theories, Methods, Individual Differences and Applications*, Erlbaum, Hillsdale, 1996.

[RIJ 96b] RIJLAARSDAM G., VAN DEN BERGH H., COUZIJN M., *Effective Teaching and Learning of Writing*, Amsterdam University Press, Amsterdam, 1996.

[SAH 08] SAHLGREN M. "The distributional hypothesis", *Rivista di Linguistica*, vol. 20, no. 1, pp. 33–53, 2008.

[SCA 87] SCARDAMALIA M., BEREITER C., "Knowledge telling and knowledge transforming in written composition", in ROSENBERG S. (ed.), *Reading, Writing and Language Learning*, Cambridge University Press, Cambridge, 1987.

[SIM 88] SIMONIN N., "An approach for creating structured text", in ZOCK M., SABAH G. (eds), *Advances in Natural Language Generation: An Interdisciplinary Perspective*, Pinter, London and Ablex, Norwood, 1988.

[TOR 99] TORRANCE M., JEFFERY G., *The Cognitive Demands of Writing*, Amsterdam University Press, Amsterdam, 1999.

[TUR 06] TURNEY P.D., "Similarity of semantic relations", *Computational Linguistics*, vol. 32, no. 3, pp. 379–416, 2006.

[WAN 06] WANG Y.W., Sentence ordering for multi-document summarization in response to multiple queries, PhD Thesis, Simon Fraser University, 2006.

[WID 04] WIDDOWS D., *Geometry of Meaning*, University of Chicago Press, Chicago, 2004.

[WIT 22] WITTGENSTEIN L., *Tractatus Logico-Philosophicus*, Kegan Paul, London, 1922.

[ZOC 15a] ZOCK M., TESFAYE D., "Automatic creation of a semantic network encoding part_of relations", *Journal of Cognitive Science*, vol. 16, no. 4, pp. 431–491, 2015.

[ZOC 15b] ZOCK M., LAPALME G., YOUSFI-MONOD M., "Learn to describe objects the way 'ordinary' people do via a web-based application", *Journal of Cognitive Science*, vol. 16, no. 2, pp. 175–193, 2015.

8

Stylistic Features Based on Sequential Rule Mining for Authorship Attribution

Authorship attribution is the task of identifying the author of a given document. Various style markers have been proposed in the literature to deal with the authorship attribution task. Frequencies of function words and Part-Of-Speech n-grams have been shown to be very reliable and effective for this task. However, despite the fact that they are state of the art, they partly rely on the invalid bag-of-words assumption, which stipulates that text is a set of independent words or segments of words. In this chapter, we present a comparative study using two different types of style markers for authorship attribution. We compare the effectiveness of using sequential rules of function words and Part-Of-Speech tags as style markers that do not rely on the bag-of-words assumption, on the one hand, and their raw frequencies, on the other hand. Our results show that the frequencies of function words and Part-Of-Speech n-grams outperform the sequential rules.

8.1. Introduction and motivation

Authorship attribution is the task of identifying the author of a given document. The authorship attribution problem can typically be formulated as follows: given a set of candidate authors for whom samples of written text are available, the task is to assign a text of unknown authorship to one of these candidate authors [STA 09].

Chapter written by Mohamed Amine BOUKHALED and Jean-Gabriel GANASCIA.

This problem has been addressed mainly as a problem of multi-class discrimination, or as a text categorization task [SEB 02]. Text categorization is a useful way to organize large document collections. Authorship attribution, as a subtask of text categorization, assumes that the categorization scheme is based on the authorial information extracted from the documents. Authorship attribution is a relatively old research field. A first scientific approach to the problem was proposed in the late 19th Century, in the work of Mendenhall in 1887, who studied the authorship of texts attributed to Bacon, Marlowe and Shakespeare. More recently, the problem of authorship attribution gained greater importance due to new applications in forensic analysis and humanities scholarship [STA 09].

To achieve high authorship attribution accuracy, we should use features that are most likely to be independent of the topic of the text. There is an agreement among different researchers that function words are the most reliable indicator of authorship. There are two main reasons for using function words in lieu of other markers. First, because of their high frequency in a written text, function words are very difficult to consciously control, which minimizes the risk of false attribution. The second is that function words, unlike content words, are more independent of the topic or the genre of the text, and hence we should not expect to find great differences of frequencies across different texts written by the same authors on different topics [CHU 07]. The Part-Of-Speech-based markers are also shown to be very effective because they partly share the advantages of function words [STA 09].

Despite the fact that function word-based markers are state-of-the-art, they basically rely on the *bag of words* assumption, which stipulates that text is a set of independent tokens. This approach completely ignores the fact that there is a syntactic structure and latent sequential information in the text. This is partly true for Part-Of-Speech n-grams as well, since they are based on the underlying assumption stipulating that text is a set of independent n-tokens' segments. De Roeck *et al*. [DER 04] have shown that frequent words, including function words, are not distributed homogeneously over a text. This provides evidence of the fact that the bag of words

assumption is invalid. In fact, critiques have been made in the field of authorship attribution charging that many works are based on invalid assumptions [RUD 97] and that researchers are focusing on attribution techniques rather than coming up with new style markers that are more precise and based on less strong assumptions.

In an effort to develop more complex yet computationally feasible stylistic features that are more linguistically motivated, Hoover [HOO 03] pointed out that exploiting the sequential information existing in the text could be a promising line of work. He proved that frequent word sequences and collocations can be used with high reliability for stylistic attribution. In another study, Quiniou *et al.* [QUI 12] showed the interest of sequential data mining methods for the stylistic analysis of large texts. They claimed that relevant and understandable patterns that may be characteristic of a specific type of text can be extracted using sequential data mining techniques.

In this line of thought, here we study the problem of authorship attribution in classic French literature. Our aim is to evaluate the effectiveness of style markers extracted using sequential data mining techniques for authorship attribution. In this contribution, we focus on extracting style markers using sequential rule mining. We compare results given by these new style markers with that of the state-of-the-art features like function words frequencies and Part-Of-Speech n-grams, and we assess whether this type of marker is sufficient for accurate identification of authors.

The rest of the chapter is organized as follows. In section 8.2, we give a theoretical overview of the computational authorship attribution process. Then, in section 8.3, we present our working hypothesis and its corresponding stylistic markers. In section 8.4, we make a projection of the sequential data mining problem in our context, and we explain how the sequential rule-based style markers are extracted. The experimental evaluation settings are presented in section 8.5 in which we describe the dataset used in the experiment, and then present the employed

classification scheme and algorithm. The results and discussions are presented in section 8.6. Finally, section 8.7 concludes the chapter.

8.2. The authorship attribution process

Authorship attribution and stylometry, which refers to the statistical analysis of literary style, have always been closely related research fields. In fact, authorship analysis relies on the notion of style and on the process of drawing conclusions about authorship information of a document by analyzing and extracting its stylometric characteristics. This assumes that the author of a document has a specific style by which he/she can be completely or partly distinguished from another author. Following this idea, current authorship attribution methods have two key steps (see Figure 8.1):

1) an indexing step based on style markers is performed on the text using some natural language processing techniques, such as Part-Of-Speech tagging, parsing and morphological analysis;

2) an identification step is applied using the indexed markers to determine the most likely authorship.

An optional feature selection step can be employed between these two key steps to determine the most relevant markers. This selection step is done by performing some statistical measures such as mutual information or Chi-square testing [HOU 06].

The identification step involves using methods that fall mainly into two categories: the first category includes methods that are based on statistical analysis, such as principle component analysis [BUR 02] or linear discriminant analysis [STA 01]; the second category includes machine learning techniques, such as simple Markov chain [KHM 01], Bayesian networks, support vector machines (SVMs) [KOP 04] and neural networks [RAM 04]. SVMs, which have been used successfully in text categorization and in other classification tasks, have been shown to be the most effective attribution method [DIE 03]. This is due to the fact that SVMs are less sensitive to irrelevant features in terms of degradation in accuracy, and enable us to handle high-dimensional data instances more efficiently.

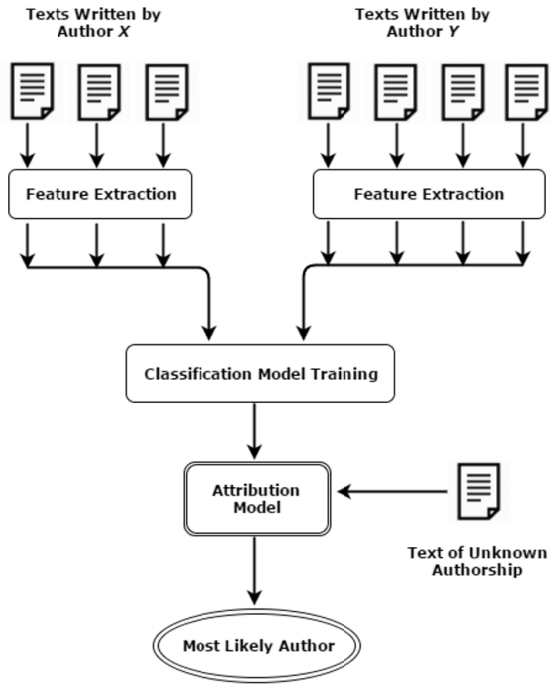


Figure 8.1. *Prototype of the process of authorship attribution process*

8.3. Stylistic features for authorship attribution

Many style markers have been used for the authorship attribution task, from early works based on features such as sentence length and vocabulary richness [YUL 44] to more recent and relevant works based on function words [HOL 01, ZHA 05], punctuation marks [BAA 02], Part-Of-Speech tags [KUK 01], parse trees [GAM 04] and character-based features [KEŠ 03]. As mentioned before, function words are shown to be a very reliable and effective indicator of authorship, and hence they are suitable to handle the task of authorship attribution or some other related tasks such as authorship verification [KOP 09]. In fact, function words have little lexical role to play, but instead they serve mostly a syntactic role by expressing grammatical relationships among words or collections of words within a sentence.

At this point, as an illustration of the idea prompted in the introduction of this chapter, we propose to explore the predictive property of stylistic

features based on sequential rule mining. Therefore, as the main experiment in our work, we study the stylistic characterization of 10 classic French authors using different stylistic features ranging from a relatively low linguistic level to a higher and more complex one. We chose to focus on the syntactic aspect of style, so as stylistic features in this experiment, we took:

– frequency of function words;

– sequential rules of function words;

– tri-gram of Part-Of-Speech tags;

– sequential rules of Part-Of-Speech tags.

From the above list, the frequencies of function words are obviously the least complex linguistic features and subsequently the least relevant and interesting to characterize the style of authors. They neither offer explicit stylistic lexical preferences, nor an explicit stylistic syntactic trait. The other stylistic features are linguistically more complex and stylistically more interesting. For instance, the sequential rules of function words can capture the differences between the periodic and loose styles. While, the sequential rules of Part-Of-Speech tags can play an alternative role in the grammatical production rules used in formal grammar [O'NE 01]; except that in this case, these rules will give insights into the syntactic choices of an author rather than describing the grammar in a general way as is done using the production rules.

What we should expect from such a configuration is that the more relevant the feature is to describe the stylistic choices of a given author, the more it is able and suitable to distinguish their own writing from that of a different author. That is to say, for a stylistic characterization based on the classification approach, we would expect the sequential rules of function words to be more effective than the frequencies of function words since they are more stylistically relevant (they are able to tell us more about the writing style of an author, they are easier to interpret at the same time, and they are not based on invalid assumptions). We would expect the sequential rules of Part-Of-Speech tags to be the more effective of the two for the same reason.

Our aim in this experiment is to test the validity of this hypothesis by evaluating the effectiveness of stylistic features presented above in the context of authorship attribution. Well, it turns out that this hypothesis is not

true, at least for the corpus that we have considered in this experiment. This can be considered as a clear argument, suggesting that less complex features, acting on a relatively low linguistic level and based on invalid assumptions, are more suitable for authorship studies from a classification point of view. Our experiment explores these issues.

8.4. Sequential data mining for stylistic analysis

Sequential data mining is a data mining subdomain introduced by Agrawal *et al*. [AGR 93], which is concerned with finding interesting characteristics and patterns in sequential databases. Sequential rule mining is one of the most important sequential data mining techniques used to extract rules describing a set of sequences. In what follows, for the sake of clarity, we will limit our definitions and annotations to those necessary to understand our experiment.

Considering a set of literals called items, denoted by $I = \{i_1, \ldots, i_n\}$, an itemset is a set of items $X \subseteq I$. A sequence S (single-item sequence) is an ordered list of items, denoted by $S = \langle i_1 \quad \ldots i_n \rangle$ where i_1 to i_n are items.

Sequence ID	Sequence
1	< a, b, d, e >
2	< b, c, e >
3	< a, b, d, e >

Table 8.1. *Sequence database SDB*

A sequence database SDB is a set of tuples (id, S), where id is the sequence identifier and S a sequence. Interesting characteristics can be extracted from such databases using sequential rules and pattern mining. A sequential rule $R : X \Rightarrow Y$ is defined as a relationship between two itemsets X and Y, such that $X \cap Y = \varnothing$. This rule can be interpreted as follows: if the itemset X occurs in a sequence, the itemset Y will occur afterward in the same sequence. Several algorithms have been developed to efficiently extract this type of rule, such as Fournier-Viger and Tseng [FOU 11]. For example, if we run this algorithm on the SDB containing the three sequences presented in Table 8.1, we will get as a result sequential rules,

such as "$a \Rightarrow d, e$" with support equal to 2, which means that this rule is respected by two sequences in the SDB (i.e. there exist two sequences of the SDB where we find the item a, and we also find d and e afterward in the same sequence).

In our study, the text is first segmented into a set of sentences, and then each sentence is mapped into two sequences: one for function words appearing in order in that sentence, and another sequence for the Part-Of-Speech tags resulting from its syntactic analysis. For example, the sentence "J'aime ma maison où j'ai grandi." will be mapped to < je,ma,où,je > as a sequence of French function words, and will be mapped to < PRO:PER, VER:pres, DET:POS, NOM, PRO:REL, PRO:PER, VER:pres, VER:pper, SENT> as a sequence of Part-Of-Speech tags. "je ⇒ où", "ma ⇒ où,je" or "DET:POS, NOM ⇒ SENT" are examples of sequential rules respected by these sequences. The whole text will produce two sequential databases, one for the function words and another for the Part-Of-Speech tags. The rules extracted in our study represent the cadence authors follow when using function words in their writings for instance. This gives us more explanatory properties about the syntactic writing style of a given author than frequencies of function words or Part-Of-Speech n-grams could offer.

8.5. Experimental setup

In this section, we present the experimental setup of our approach. We first describe the dataset used in the experiment, and then present the classification scheme and algorithm employed for this experiment. The results and discussion are presented in the next section.

8.5.1. *Dataset*

To test the effectiveness of sequential rules over Part-Of-Speech tags and function words for authorship attribution, we used texts written by Balzac, Dumas, France, Gautier, Hugo, Maupassant, Proust, Sand, Sue and Zola. This choice was motivated by our special interest in studying the classic French literature of the 19th Century, and the availability of electronic texts from these authors on the Gutenberg project website[1] and in the Gallica

1 http://www.gutenberg.org/

electronic library[2]. Our choice of authors was also affected by the fact that we want to cover the most important writing styles and trends from this period. For each of the 10 authors mentioned above, we collected four novels, so that the total number of novels is 40. The next step was to divide these novels into smaller pieces of texts in order to have enough data instances to train the attribution algorithm. Researchers working on authorship attribution on literature data have been using different dividing strategies. For example, Hoover [HOO 03] decided to take just the first 10,000 words of each novel as a single text, while Argamon and Levitan [ARG 05] treated each chapter of each book as a separate text. In our experiment, we chose to slice novels by the size of the smallest one in the collection in terms of the number of sentences. This choice respects the condition proposed by Eder [EDE 13] that specifies the smallest reasonable text size to achieve good attribution; more information about the dataset used in the experiment is presented in Table 8.2.

Author Name	# of words	# of texts
Balzac, Honoré de	548778	20
Dumas, Alexandre	320263	26
France, Anatole	218499	21
Gautier, Théophile	325849	19
Hugo, Victor	584502	39
Maupassant, Guy de	186598	20
Proust, Marcel	700748	38
Sand, George	560365	51
Sue, Eugène	1076843	60
Zola, Émile	581613	67

Table 8.2. *Statistics for the dataset used in our experiment*

8.5.2. *Classification scheme*

In the current approach, each text was segmented into a set of sentences (sequences) based on splitting done using the punctuation marks of the set {'.', '!', '?', ':', '…'}, then the corpus was Part-Of-Speech tagged and function words were extracted. The algorithm described in Fournier-Viger and Tseng [FOU 11] was then used to extract sequential and association

2 http://gallica.bnf.fr/

rules over the function words and the Part-Of-Speech tag sequences from each text. These rules will help us gather not only sequential information from the data, but also structural information, due to the fact that a text characterized by long sentences will result in more frequencies of the rules.

Each text is then represented as a vector R_K of frequencies of occurrence of rules, such that $R_K = \{r_1, r_2, \ldots, r_K\}$ is the ordered set, by decreasing normalized frequency of occurrence of the top-K rules in terms of support in the training set. Each text is also represented by a vector of normalized frequencies of occurrence of function words and Part-Of-Speech tag 3-grams. The normalization of the vector of frequency representing a given text was done by the size of the text. Our aim is first to compare the classification performance of the top-K function word sequential rules (SR) with the function words frequencies. Second, to compare the classification performance of the top-K sequential rules of Part-Of-Speech tag with the 3-gram frequencies.

Given the classification scheme described above, we used SVMs classifier to derive a discriminative linear model from our data. To get a reasonable estimation of the expected generalization performance, we used 5-fold cross-validation. The dataset was split into five equal subsets; the classification was done five times by taking four subsets for training each time and leaving out the last one for testing. The overall classification performance is taken as the average performance over these five runs. In order to evaluate the attribution performance, we used the common measures used to evaluate supervised classification performance: we have calculated precision (P), recall (R) and F-measure F_β, where TP stands for true positive, TN for true negative, FP for false positive and FN for false negative:

$$P = \frac{TP}{TP + FP} \qquad [8.1]$$

$$R = \frac{TP}{TP + FN} \qquad [8.2]$$

$$F_\beta = \frac{\left(1 + \beta^2\right) RP}{\left(\beta^2 R\right) + P} \qquad [8.3]$$

We consider that precision and recall have the same weight, and hence we set β equal to 1.

8.6. Results and discussion

The results of measuring the attribution performance for the different feature sets presented in our experiment setup are summarized in Table 8.3 for features derived from function words, and in Table 8.4 for those derived from Part-Of-Speech tags. These results show, in general, a better performance when using function words and Part-Of-Speech tag 3-gram frequencies, which achieved a nearly perfect attribution, over features based on sequential rules for our corpus.

Our study here shows that the SVMs classifier combined with features extracted using sequential data mining techniques can achieve a high attribution performance (e.g. F1 = 0.939 for Top 300 FW-SR). Until a certain limit, adding more rules increases the attribution performance (e.g. F1 = 0.733 for Top 100 POS-SR compared with F1 = 0.880 for Top 800 POS-SR).

Contrary to our hypothesis, function word frequency features, which fall under the bag-of-word assumption, known to be blind to sequential information, outperform features extracted using the sequential rule mining technique. The same thing can be said for the Part-Of-Speech tag 3-grams.

Feature set	P	R	F_1
Top 100 FW-SR	0.901	0.886	0.893
Top 200 FW-SR	0.942	0.933	0.937
Top 300 FW-SR	0.940	0.939	0.939
FW frequencies	**0.990**	**0.988**	**0.988**

Table 8.3. *Five-fold cross-validation for our dataset. SR refers to sequential rules and FW refers to function words*

By taking a closer look at the sequential rules extracted from the Part-Of-Speech tag sequences, we found that these rules, especially the most frequent ones, are more likely to be language-grammar dependent (e.g. ADJ NC,PONCT with sup = 63,569 and DET,NC,P ⇒ ADJ with sup = 63,370).

To reduce this effect, we added a $TF-IDF$-like heuristic that measures the overall discriminative power of each sequential rule. The $TF-IDF$-like weight of a sequential rule R_i present in a text t is calculated as follows:

$$TF-IDF_t(R_i)=\left(1+supp_t(R_i)\right)*\log\left(\frac{N}{N_t}\right) \quad [8.4]$$

where $supp_t(R_i)$ is the support of the rule R_i in the text t, N is the total support of all rules in the corpus and N_t is the total support of all rules in the text t.

Results given by this $TF-IDF$ weighting in Table 8.5 are better than the original ones, but they still cannot reach the performance given by the state-of-the-art style markers. This suggests that in future studies, we should add an adequate feature selection method that will filter the rules to capture the most relevant ones.

By analyzing the individual attribution performance for each author separately, we notice a significant variance between the attribution performance of one author and that of another (e.g. $F_1 = 1$ for Proust compared with $F_1 = 0.673$ for Dumas); some individual results are presented in Table 8.5. This particularity is due to the fact that some authors have more characterizing style than others in the works used for the experiment. This property can be clearly visualized by carrying out the principal components analysis (see Figure 8.2) on the 40 books used in the dataset.

Feature set	P	R	F_1
Top 200 POS-SR	0.72	0.70	0.71
Top 300 POS-SR	0.83	0.81	0.82
Top 400 POS-SR	0.84	0.83	0.83
Top 500 POS-SR	0.85	0.84	0.84
Top 600 POS-SR	0.87	0.85	0.86
Top 700 POS-SR	0.88	0.86	0.87
Top 800 POS-SR	0.88	0.87	0.88
POS 3-gram frequencies	**0.99**	**0.99**	**0.99**

Table 8.4. *Five-fold cross-validation results for our dataset. SR refers to sequential rules and POS refers to Part-Of-Speech*

Feature set	P^*	R^*	F^*_1
Top 200 POS-SR	0.82	0.79	0.81
Top 300 POS-SR	0.86	0.84	0.85
Top 400 POS-SR	0.87	0.86	0.87
Top 500 POS-SR	0.89	0.88	0.88
Top 600 POS-SR	0.89	0.88	0.88
Top 700 POS-SR	0.91	0.90	0.90
Top 800 POS-SR	**0.92**	**0.91**	**0.91**

Table 8.5. *Five-fold cross-validation results given by considering the TF-IDF-like weighting for our dataset. SR refers to sequential rules and POS refers to Part-Of-Speech*

Even if these results are in line with previous works that claimed that bag-of-words-based features are more relevant than sequence-based features for stylistic attribution [ARG 05], they show that style markers extracted using sequential rule mining techniques can be valuable for authorship attribution. We believe that our results open the door to a promising line of research by integrating and using sequential data mining techniques to extract more linguistically motivated style markers for computational, stylistic and authorship attribution.

Author Name	P	R	F_1
Balzac	0.88	0.75	0.80
Dumas	0.65	0.69	0.67
France	0.92	0.96	0.93
Gautier	0.95	0.85	0.89
Hugo	0.88	0.95	0.91
Maupassant	1.00	0.85	0.91
Proust	1.00	1.00	1.00
Sand	0.92	0.90	0.91
Sue	0.86	0.86	0.86
Zola	0.98	1.00	0.99

Table 8.6. *Individual 5-fold cross-validation results for each author evaluated for the Top 700 Part-Of-Speech tag sequential rules*

Actually, despite the fact that function words are not very relevant features to describe the stylistic characterization, they are a reliable indicator of

authorship. Owing to their high frequency in a written text, function words are very difficult to consciously and voluntarily control, which makes them a more inherent trait and consequently minimizes the risk of false attribution. Moreover, unlike content words, they are more independent of the topic or the genre of the text, and therefore we should not expect to find great differences of frequencies across different texts written by the same authors on different topics [CHU 07]. Yet, they basically rely on the bag-of-words assumption, which stipulates that text is a set of independent words.

As we have seen, it turns out that the hypothesis, stated as a basis for the experiment, is not true, at least for the corpus that we have considered in this experiment. This can be considered as a clear argument, suggesting that complex features such as sequential rules are not suitable for authorship attribution studies. In fact, there is a difference between the characterizing ability of a stylistic feature, on the one hand, and its discriminant power, on the other. The most relevant and suitable stylistic features to perform a discriminant task such as stylistic classification are the ones that operate on the low linguistic levels as function words do. These are subsequently more difficult to linguistically interpret and understand and do not necessarily enhance the knowledge concerning the style of the text from which they were extracted.

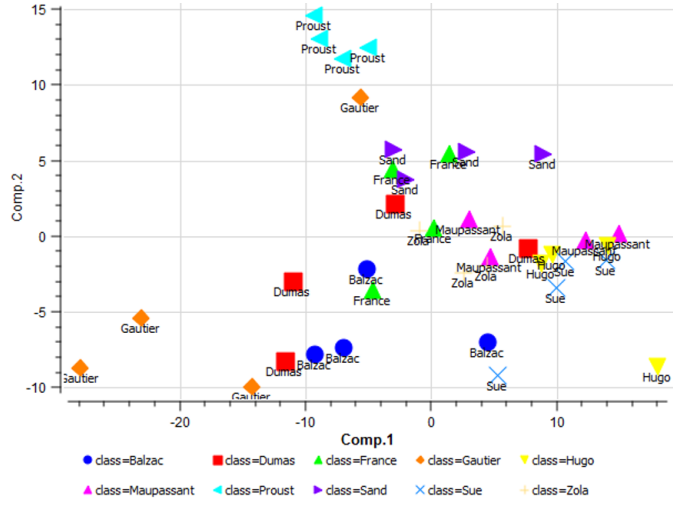


Figure 8.2. *Principal components analysis of the 40 books (four books per author) in the dataset, Top 200 SR analyzed. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

8.7. Conclusion

In this chapter, we have presented a first study on using style markers extracted using sequential data mining techniques for authorship attribution. We have considered extracting linguistically motivated markers using a sequential rule mining technique based on function word and Part-Of-Speech tags. To evaluate the effectiveness of these markers, we conducted experiments on a classic French corpus. Our preliminary results show that sequential rules can achieve a high attribution performance that can reach an F_1 score of 93%. Yet, they still do not outperform low-level features, such as frequencies of function words.

Based on the current study, we have identified several future research directions. First, we will explore the effectiveness of using probabilistic heuristics to find a minimal feature set that still allows good attribution performance, which would be very helpful for stylistic and literary analysis. Second, this study will be expanded to include sequential patterns (n-gram with gaps) as style markers. Third, we intend to experiment with this new type of style markers for other languages and text sizes using standard corpora employed in the wider field.

8.8. Bibliography

[AGR 93] AGRAWAL R., IMIELIŃSKI T., SWAMI A., "Mining association rules between sets of items in large databases", *ACM SIGMOD*, vol. 22, no. 2, pp. 207–216, 1993.

[ARG 05] ARGAMON S., LEVITAN S., "Measuring the usefulness of function words for authorship attribution", *Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, pp. 3–7, 2005.

[BAA 02] BAAYEN H., VAN HALTEREN H., NEIJT A. *et al*., "An experiment in authorship attribution", *Proceedings of 6th International Conference on the Statistical Analysis of Textual Data*, pp. 29–37, 2002.

[BUR 02] BURROWS J., "'Delta': a measure of stylistic difference and a guide to likely authorship", *Literary and Linguistic Computing*, vol. 17, no. 3, pp. 267–287, 2002.

[CHU 07] CHUNG C., PENNEBAKER J., "The psychological functions of function words", *Social Communication*, pp. 343–359, 2007.

[DER 04] DE ROECK A., SARKAR A., GARTHWAITE P., "Defeating the homogeneity assumption", *Proceedings of 7th International Conference on the Statistical Analysis of Textual Data*, pp. 282–294, 2004.

[DIE 03] DIEDERICH J., KINDERMANN J., LEOPOLD E. *et al.*, "Authorship attribution with support vector machines", *Applied Intelligence*, vol. 19, pp. 109–123, 2003.

[EDE 13] EDER M., "Does size matter? Authorship attribution, small samples, big problem", *Digital Scholarship in the Humanities*, 2013.

[FOU 11] FOURNIER-VIGER P., TSENG V., "Mining top-k sequential rules", *Advanced Data Mining and Applications*, Springer, pp. 180–194, 2011.

[GAM 04] GAMON M., "Linguistic correlates of style: authorship classification with deep linguistic analysis features", *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 611–617, 2004.

[HOL 01] HOLMES D., ROBERTSON M., PAEZ R., "Stephen Crane and the New-York Tribune: a case study in traditional and non-traditional authorship attribution", *Computers and the Humanities*, vol. 35, no. 3, pp. 315–331, 2001.

[HOO 03] HOOVER D., "Frequent collocations and authorial style", *Literary and Linguistic Computing*, vol. 18, no. 3, pp. 261–286, 2003.

[HOU 06] HOUVARDAS J., STAMATATOS E., "N-gram feature selection for authorship identification", *Artificial Intelligence: Methodology, Systems, and Applications,* Springer, pp. 77–86, 2006.

[KEŠ 03] KEŠELJ V., PENG F., CERCONE N. *et al.*, "N-gram-based author profiles for authorship attribution", *Proceedings of the Conference Pacific Association for Computational Linguistics*, PACLING, vol. 3, pp. 255–264, 2003.

[KHM 01] KHMELEV D., TWEEDIE F., "Using Markov chains for identification of writer", *Literary and Linguistic Computing*, vol. 16, no. 3, pp. 299–307, 2001.

[KOP 04] KOPPEL M., SCHLER J., "Authorship verification as a one-class classification problem", *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 62–67, 2004.

[KOP 09] KOPPEL M., SCHLER J., ARGAMON S., "Computational methods in authorship attribution", *Journal of the American Society for Information Science and Technology*, vol. 60, no. 1, pp. 9–26, 2009.

[KUK 01] KUKUSHKINA O., POLIKARPOV A., KHMELEV D., "Using literal and grammatical statistics for authorship attribution", *Problems of Information Transmission*, vol. 37, no. 2, pp. 172–184, 2001.

[O'NE 01] O'NEILL M., RYAN C., "Grammatical evolution", *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 349–358, 2001.

[QUI 12] QUINIOU S., CELLIER P., CHARNOIS T. *et al*., "What about sequential data mining techniques to identify linguistic patterns for stylistics?", *Computational Linguistics and Intelligent Text Processing*, Springer, pp. 166–177, 2012.

[RAM 04] RAMYAA C.H., RASHEED K., "Using machine learning techniques for stylometry", *Proceedings of International Conference on Machine Learning*, 2004.

[RUD 97] RUDMAN J., "The state of authorship attribution studies: some problems and solutions", *Computers and the Humanities*, vol. 34, no. 4, pp. 351–365, 1997.

[SEB 02] SEBASTIANI F., "Machine learning in automated text categorization", *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[STA 01] STAMATATOS E., FAKOTAKIS N., KOKKINAKIS G., "Computer-based authorship attribution without lexical measures", *Computers and the Humanities*, vol. 35, pp. 193–214, 2001.

[STA 09] STAMATATOS E., "A survey of modern authorship attribution methods", *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[YUL 44] YULE G., *The Statistical Study of Literary Vocabulary*, CUP Archive, 1944.

[ZHA 05] ZHAO Y., ZOBEL J., "Effective and scalable authorship attribution using function words", *Information Retrieval Technology*, Springer, pp. 174–189, 2005.

This page intentionally left blank

9

A Parallel, Cognition-oriented Fundamental Frequency Estimation Algorithm

9.1. Introduction

The fundamental frequency F0 plays an important role in human speech perception and is used in all fields of speech research. For instance, humans identify emotional states based on a few features, one of which is F0 [ROD 11]. For speech synthesis, accurate estimates of F0 are a prerequisite for prosody control in concatenative speech synthesis [EWE 10].

Fundamental frequency detection has been an active field of research for more than 40 years. Early methods used the autocorrelation function and inverse filtering techniques [MAR 72, RAB 76]. In most of these approaches, threshold values are used to decide whether a frame is assumed to be voiced or unvoiced. More advanced methods incorporate a dynamic programming stage to calculate the F0 contour based on frame-level F0 estimates gained from either a conditioned linear prediction residual or a normalized cross correlation function [SEC 83, TAL 95]. The normalized cross correlation-based RAPT algorithm is also known as getf0. Praat's well-known pitch detection algorithm calculates cross-correlation or autocorrelation functions and considers local maxima as F0 hypotheses

Chapter written by Ulrike GLAVITSCH.

[BOE 01]. The fundamental frequency estimator for speech and music YIN with no upper limit on the frequency search range uses the autocorrelation function and a number of modifications to prevent errors [DEC 02]. In the last decade, techniques like pitch-scaled harmonic filtering (PSHF), non-negative matrix factorization (NMF) as well as time-domain probabilistic approaches have been proposed for F0 estimation [ACH 05, ROA 07, SHA 05, PEH 11]. In SAFE, F0 estimates are inferred from prominent signal-to-noise ratio (SNR) peaks in the speech spectra [CHU 12]. Pitch and probability-of-voicing estimates gained from a highly modified version of the getf0 (RAPT) algorithm are used in an automatic speech recognition system for tonal languages [GHA 14]. These recent methods achieve low error rates and high accuracies but at a high computational cost – either at run-time or during model training. These calculative approaches generally disregard the principles of human cognition and the question is whether F0 estimation can be performed equally well or better by considering them.

In this chapter, we propose an F0 estimation algorithm based on the elementary appearance and inherent structure of the human speech signal. A period, i.e. the inverse of F0, is primarily defined as the time distance between two maximum and two minimum peaks, and we use the same term to refer to the speech section between two such peaks. Human speech is a sequence of alternating speech and pause segments. Speech segments are word flows uttered in one breath of air. The speech segments are usually much longer than the pause segments. In speech segments, we distinguish voiced and unvoiced parts. The speech signal is periodic in voiced regions, whereas it is aperiodic in unvoiced regions. The voiced regions can be further divided into stable and unstable intervals [GLA 15]. Stable intervals show a quasi-constant energy or a quasi-flat envelope, whereas unstable intervals exhibit significant energy rises or decays. On stable intervals, the F0 periods are mostly regular, i.e. the sequence of maximum or minimum peaks is more or less equidistant, whereas the F0 periods in unstable regions are often shortened, elongated, doubled, or may show little similarity with their neighboring periods. Speech signals are highly variable and such special cases occur relatively often. Thus, it makes sense to compute F0 estimates in stable intervals first and use this knowledge to find F0 of unstable intervals in a second step. The F0 estimation method for stable intervals is straightforward as regular F0 periods are expected. The F0

estimation approach for unstable intervals computes variants of possible F0 continuation sequences and evaluates them for highest plausibility. The variants reflect the regular and all the irregular period cases and are calculated using a peak look-ahead strategy. We denote this F0 estimation method for unstable intervals as F0 propagation, since it computes and verifies F0 estimates by considering previously computed ones.

It turns out that the whole F0 estimation can be performed in parallel on the different speech segments of a recording. The speech segments can be considered as separable units of speech that can be treated as computationally independent entities.

We consider the proposed algorithm as cognition oriented inasmuch as it incorporates several principles of human cognition. First, human hearing is also a two-stage process. The inner ear performs a spectral analysis of a speech section, i.e. different frequencies excite different locations along the basilar membrane and as a result different neurons with characteristic frequencies [MOO 08]. This spectral analysis delivers the fundamental frequency and the harmonics. The brain, however, then checks the information delivered by the neurons, interpolating and correcting it where necessary. Our proposed F0 estimation algorithm performs in a similar way, in that the F0 propagation step proceeds from regions with reliable F0 estimates to those where F0 is not clearly known yet. We observed that F0 is very reliably estimated on high-energy stable intervals, which typically represent vowels. Thus, we always compute F0 for unstable intervals by propagation from high-energy stable intervals to lower energy regions. Second, we have adopted the hypothesis-testing principle of human thinking for generating variants of possible F0 sequences and testing them for the detection of F0 in unstable intervals [KAH 11]. Next, human cognition uses context to decide a situation. For instance, in speech perception humans bear the left and right context of a word in mind if its meaning is ambiguous. In an analogous way, our algorithm looks two or three peaks ahead to find the next valid maximum or minimum peak for a given F0 hypothesis. Special cases in unstable intervals can very rarely be disambiguated by just looking a single peak ahead. Finally, performing the tasks of the F0 estimation algorithm in parallel on different speech segments is also adopted from human cognition. The human brain is able to process a huge number of tasks in parallel.

The resulting algorithm is very efficient, thoroughly extensible, easy to understand and has been evaluated on a clean speech database. Recognition rates are better than those of a reference method that uses cross-correlation functions and dynamic programming. In addition, our algorithm structures the speech signal in spoken and pause segments, voiced and unvoiced regions, and stable and unstable intervals. This structure may be useful for further speech processing, such as automatic text-to-speech alignment, automatic speech or speaker recognition.

9.2. Segmentation of the speech signal

As mentioned in the introduction, a speech signal consists of speech units separated by pauses. The speech units contain voiced and unvoiced regions and on the voiced parts, we distinguish stable and unstable intervals. The algorithms and criteria to detect these different structures are described in the following sections.

9.2.1. *Speech and pause segments*

We use the algorithm to determine the endpoints of isolated utterances by Rabiner and extend it by a heuristics to find the pauses between the spoken segments in a speech signal [RAB 75]. We refer to this combined algorithm as a pause-finding algorithm. Rabiner's algorithm decides whether a signal frame, i.e. a small 10 ms long section of the signal, is characterized as speech or pause based on its energy and the silence energy. The silence energy is the mean energy of an interval that contains silence or signal noise. The silence or noise in our algorithm is expected at the beginning of the speech signal. Users may configure the length over which the silence energy is computed; the default value is 100 ms. First, the pause-finding algorithm calculates an initial segment list, where each segment is characterized by its start and end sample positions and the segment type – either SPEECH or PAUSE. Second, the algorithm merges pause segments that are too short with their neighboring speech segments. In a similar way, it merges speech segments that are too short with their neighboring pause segments. Some of the segments in the initial segment list are too short to form a true speech or pause segment. For instance, a glottal stop before a plosive or a

low-energy speech segment is often identified as a pause segment. The minimum lengths of both pause and speech segments are configurable. Finally, the algorithm extends the speech segments by a certain small length. This is necessary since the ends of speech segments may be low-energy phonemes. These phonemes are automatically included by extending the speech segments by a configurable length. The pause-finding algorithm consists of six steps that we present in the following:

1) *Energies, peak energy and silence energy*: the energies E(k) are computed at discrete points k every 10 ms each over a window of 10 ms in the speech signal, k = 0, …, n – 1. The peak energy Emax is the maximum energy of all energies E(k). Emin is the mean energy of the initial silence that is supposed to occur at the beginning of the speech signal.

2) *Threshold ITL for speech/pause decision*: the threshold ITL is computed as in [RAB 75]:

$$I1 = 0.03 * (Emax - Emin) + Emin \quad [9.1]$$

$$I2 = 4 * Emin \quad [9.2]$$

$$ITL = \min(I1, I2) \quad [9.3]$$

3) *Initial segment list*: each frame is classified as either speech or pause frame, comparing its energy with ITL. It is a speech frame if its energy is larger than ITL, a pause frame otherwise. Consecutive speech frames form a speech segment, and consecutive pause frames form a pause segment of the initial segment list.

4) *Merging of too short segments of type PAUSE*: pause segments shorter than the configurable minimum pause length are merged with their neighboring speech segments. The default minimum pause length is 200 ms.

5) *Merging of too short segments of type SPEECH*: speech segments shorter than the configurable minimum speech length are merged with their neighboring pause segments. The default minimum speech length is 150 ms.

6) *Extension of segments of type SPEECH*: all segments of type SPEECH are extended by the length given by the configurable maximum speech segment extension (default value 50 ms). At the same time, the pause segments to the left and right of each speech segments are reduced by that amount.

Figure 9.1 shows the result of the pause-finding algorithm for a speech recording, where a female speaker reads the beginning of the story "The north wind and the sun" of the Keele pitch database [PLA 95].

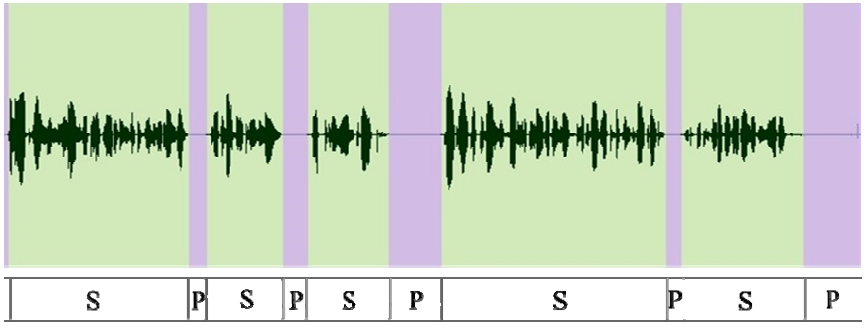


Figure 9.1. *Segmentation of a speech signal into speech (S) and pause (P) segments. For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

9.2.2. *Voiced and unvoiced regions*

Voiced and unvoiced regions are detected in the speech segments only. The pause segments are unvoiced by default. To detect voiced and unvoiced regions, the speech segment is again subdivided into a sequence of frames. However, the frames are longer than that for the pause-finding algorithms, i.e. 20 ms, and they overlap by half of the frame length, namely 10 ms. We define a frame to be voiced if its mean energy exceeds a certain threshold, the absolute height of the frame's maximum or minimum peak is above a given level and the number of counted zero crossings in the frame is lower than a certain number. A voiced region is a sequence of voiced frames and similarly, an unvoiced region contains only unvoiced frames.

A zero crossing is the location in the speech signal where there is change from a positive sample value to a negative value or vice versa. Voiced regions such as vowels, nasals, etc. exhibit a low number of zero crossings, whereas unvoiced regions, e.g. fricatives, usually have a rather high number of zero crossings.

For computation of the mean energy of a frame, we use a more elaborate method than the standard approach, i.e. computing the sum of the squares of the frame's samples and dividing it by the frame length. This standard approach is not precise if the F0 of a frame is not an integer multiple of the frame length [GLA 15]. It may falsify the voiced/unvoiced decision and the stable/unstable classification of a frame in a later step (see section 9.2.3) that is based on the mean energy, too. However, as the period of a frame – the inverse of F0 – is not known at this stage of processing, we compute the mean energy on a scale of window lengths, each of which corresponds to a different period length. An optimization step then finds the best window length for each frame. This procedure is similar to pitch-scaled harmonic filtering (PSHF) [ROA 07], where an optimal window length is calculated for finding harmonic and non-harmonic spectra. The window lengths are selected such that periods of F0 between 50 and 500 Hz roughly fit in one of the selected windows a small number of times at least. The selected window lengths correspond to fundamental frequencies of 50, 55, 60,…, 95 Hz. Each window length is centered around a frame's center position. The optimal window length is the one where the mean energies of a small number of frames around the frame's middle position show the least variation [GLA 15].

9.2.3. *Stable and unstable intervals*

We further segment voiced regions into stable and unstable intervals. As mentioned in the introduction, stable intervals have a quasi-constant energy, whereas in unstable intervals, the energy rises or falls significantly. Given the mean energy of a frame as computed in section 9.2.2, a frame is defined as stable in the following way: its mean energy must not deviate by more than 50% from the mean energy of the previous frame and also not by more than 50% from the mean energy of the next frame. By setting the threshold for the relative mean energy difference to 50%, we allow some tolerance for the energy differences between frames of a stable interval. This is justified since speech signals show high variations.

Figure 9.2 shows a voiced region of a speech segment with three stable intervals S_1, S_2 and S_3. The figure also depicts the series of overlapping speech frames for processing.

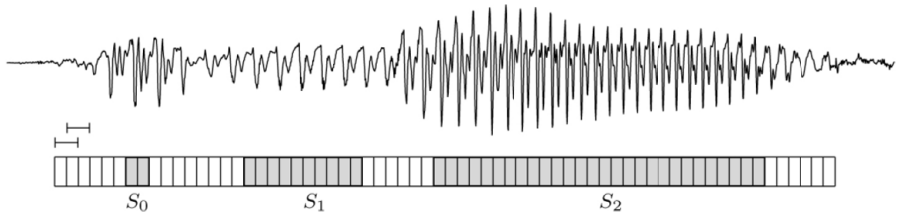


Figure 9.2. *Overlapping frames of a voiced region of a speech segment that contains the words "the north" uttered by a male speaker. Three stable intervals S_1, S_2 and S_3 of lengths 1, 9 and 27 are identified.*

9.3. F0 estimation for stable intervals

The F0 estimation method for stable intervals finds a quadruple of signal peaks $P = (p_L, p_0, p_1, p_R)$ of either maximum or minimum peaks p_i, i = L, 0, 1, R, such that the center position of the frame is between p_0 and p_1. A peak is defined as either a local minimum or a local maximum in the sequence of signal samples. For each peak, p_k, k = 0, …, n – 1, in a speech segment, we maintain a triple of values (x_k, y_k, c_k), where x_k and y_k are the peak coordinates and c_k is the peak classification – either a minimum or a maximum. The F0 estimate is the inverse of the mean of the period lengths found in P, i.e. the mean of the distances between peaks p_L and p_0, p_0 and p_1 as well as p_1 and p_R. The tuple P is selected among a series of possible candidate tuples according to a similarity score. Furthermore, it is checked whether the peak tuple is not a multiple of the supposedly true F0 period, otherwise a different peak tuple is selected. In the following, we describe the algorithm to find such a peak tuple P for each stable frame.

We start by finding the peak in the frame that has the highest absolute value. We then look for candidate peaks that have a similar absolute height and whose distance from the highest peak is within the permissible range of period lengths. The search for candidate peaks is performed in the direction of the center position of the frame. Given a peak pair p_0 and p_1 – one of them with the highest absolute value and a candidate peak – the algorithm looks for peaks to the left and the right to complete the quadruple. We select those peaks with the highest absolute values in about the same distance to the left and to the right of p_0 and p_1 as the distance between the two peaks. The peak quadruple may reduce to a triple peak sequence if such a peak at one side of

the middle peak pair cannot be found. Each such candidate peak quadruple or peak triple is scored and the tuple with the highest score is selected as the tentatively best candidate.

The proposed score measures the equality of peak distances and absolute peak heights of a peak tuple. The score s for peak tuple $P = (p_L, p_0, p_1, p_R)$ is the product of partial scores s_x and s_y. The value s_x measures the equality of the peak intervals, whereas s_y is a measure for the similarity of the absolute peak heights. The partial score s_x is defined as 1 – a, where a is the root of the mean squared relative differences between the peak distances at the edges and the distance between the middle peaks. The partial score s_y is given as 1 – b, where b is the root of the mean squared difference of the absolute peak heights from the maximum absolute peak height in the given peak tuple. The equations below show how the score s is computed for a peak quadruple in detail. The formulas are easily adapted for tuples with only three peaks:

$$s = s_x s_y \qquad [9.4]$$

The partial score s_x is defined as follows:

$$s_x = 1 - a \qquad [9.5]$$

$$a = \sqrt{(b_0^2 + b_1^2)/2} \qquad [9.6]$$

$$b_0 = \frac{d_1 - d_0}{d_1}, b_1 = \frac{d_1 - d_2}{d_1} \qquad [9.7]$$

$$d_0 = x_0 - x_L, d_1 = x_1 - x_0, d_2 = x_R - x_1 \qquad [9.8]$$

The value x_i, i = L, 0, 1, R, refers to the x-coordinate of peak p_i as mentioned above.

The partial score s_y is given by:

$$s_y = 1 - b \qquad [9.9]$$

$$b = \sqrt{\frac{1}{4}(g_L^2 + g_0^2 + g_1^2 + g_R^2)} \quad [9.10]$$

$$g_i = (|y_i| - y_{\max}) / y_{\max}, i = L, 0, 1, R \quad [9.11]$$

$$y_{\max} = \max(|y_i|, i = L, 0, 1, R) \quad [9.12]$$

The value y_i denotes the peak height of peak p_i, i = L, 0, 1, R. The score s delivers exactly 1 if the peak heights and peak intervals are equal and less than 1 if they differ.

The peak tuple with the highest score may be a multiple of the true period. Thus, we check for the existence of equidistant partial peaks within the peak pair p_0 and p_1. Such partial peaks must have about the same absolute height as the original candidate peaks p_0 and p_1. If such partial peaks on both sides of the x-axis are found, we look for a candidate peak tuple with the partial peak distance and install it as the current best candidate.

Next, we find the peak tuple in the center of the frame that has the same period length as the best candidate tuple. This is achieved by looking for peaks to either the left or the right side of the best candidate in the distance of the period length until a peak tuple is found, where the frame's center is between the two middle peaks p_0 and p_1.

Finally, we detect sequences of roughly equal F0 estimates in a stable interval. These sequences are referred to as equal sections. The F0 estimates of the frames in an equal section must not deviate from the mean F0 in the equal section by more than a given percentage that is currently set to 10%. The longest such equal section with a minimum length of 3 is set as *the* equal section of the stable interval. The remaining equal sections are maintained in a list and may be used during F0 propagation (see section 9.4).

9.4. F0 propagation

The F0 propagation is the second major stage of the proposed F0 estimation algorithm. Its purpose is to calculate and check F0 estimates in regions where no reliable F0 estimates exist. This mainly affects unstable

intervals and also portions of stable intervals where, for example, the F0 estimates do not belong to an equal section. The main idea is that the F0 propagation starts at the stable interval with the highest energy from where it proceeds to the regions to both its left and right side. It always progresses from higher energy to lower energy regions. Once a local energy minimum is reached, it continues with the next stable interval in propagation direction, i.e. a local energy maximum. For the verification and correction of calculated F0 estimates, we developed a peak propagation procedure that computes the most plausible peak continuation sequence given the peak tuple of the previous frame. The most plausible peak continuation sequence is found by considering several variants of peak sequences that reflect the regular and irregular period cases. In the following, we describe the control flow of the F0 propagation and explain the particular peak propagation procedure.

9.4.1. *Control flow*

The propagation of F0 estimates is performed separately for each voiced region. The first step in this procedure is the definition of the propagation order and the propagation end points. The propagation starts with the stable interval that contains the frame with the highest mean energy in the equal section. From this equal section, the propagation flows first to the left side and then to the right side. For each stable interval containing an equal section, we define the right and the left propagation end points. They are the start and end frames of the voiced region if there is only one stable interval in the voiced region. The propagation end points are the local energy minimum frame and its direct neighbor if there is a local energy minimum between two stable intervals S_1 and S_2. They are the start frame and the preceding frame of S_2 if there is no local energy minimum between S_1 and S_2 and S_2 has lower energy than S_1. In a similar way, the propagation end points are the end frame and its successor frame of S_1 if there is no local energy minimum between S_1 and S_2 and S_1 has lower energy than S_2. Figure 9.3 shows the propagation directions, order and end points of a voiced segment that contains two stable intervals with equal sections E_0 and E_1. For simplicity, the stable interval that contains E_0 and the stable interval that contains E_1 are not shown in the figure.

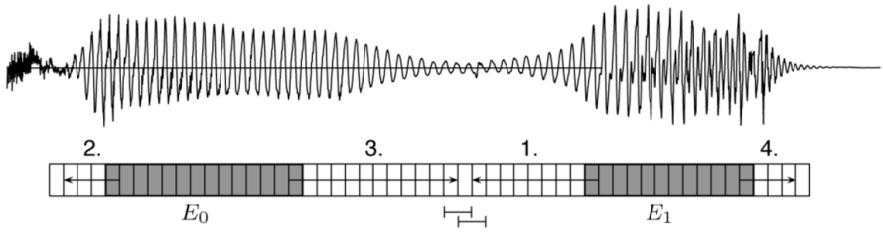


Figure 9.3. *Propagation order, directions and end points of a voiced region. Propagation start and end points are marked at the center positions of frames. Propagation starts from equal section E_1 as it has frames with higher energies than E_0*

Next, we compute candidate F0 values for the unstable frames using the method presented in section 9.3, but with a restricted range of tolerable F0. We calculate the mean F0 of the equal section where the propagation starts. The lower and upper bounds of the tolerable F0 range are an octave lower than this mean F0 value and two-thirds of an octave higher than it. In contrast to the F0 estimation method for stable intervals, the check for multiple periods is omitted since it would hardly work in unstable intervals with potentially strongly varying peak heights.

The main part of the F0 propagation stage is to check whether the F0 estimate of a frame is in accordance with the F0 of its previous frame and if not, to perform a peak propagation step (see section 9.4.2) to find the most plausible peak continuation sequence. The frame's actual F0 is derived from the detected peak continuation sequence. The peak continuation sequence may be regular but may also contain elongated or shortened periods or octave jumps. As soon as a propagation end point is reached, we check whether the mean F0 of the equal section of the next stable interval is similar to the mean F0 of the most recently calculated values. Propagation continues normally from the next stable interval if this condition holds. Otherwise, the list of equal sections in a stable interval is inspected for a better fitting equal section and the algorithm uses this as the new propagation starting point if such an equal section is found.

9.4.2. *Peak propagation*

The peak propagation step computes a set of peak sequence variants that may follow the peak tuple of the previous frame and evaluates each of them for plausibility. Each peak sequence is computed by a look-ahead strategy for the next peak. In general, we look two peaks ahead before deciding on the next one.

The following peak sequence variants are considered:

– V1 (regular case): the peaks continue at about the same distance as the peaks in the previous frame;

– V2 (elongated periods): the periods are elongated and the peak distances become larger;

– V3 (octave jump down): the peaks follow at double distance as in the previous frame;

– V4 (octave jump up): the peaks follow at half the distance as in the previous frame.

The peak sequence variants are computed depending on the octave jump state of the previous frame. The octave jump state is maintained for each frame and its default value is “none”. There are two additional values, “down” and “up”, for the state of F0 that is an octave higher than normal, and the state of an F0 estimate that has fallen by an octave. Variant V2 is used to detect extended periods that may not be captured by V1. V3 is necessary to test the case of a sudden octave jump down but is only calculated in the case of an octave jump “none”. V4 is considered only in an octave jump down state to check whether such a phase has ended. Currently, our algorithm detects neither repeated octave jumps down nor a sudden octave jump up, but the recognition of these cases can be implemented in the future.

For each of the variants V1–V4, we define interval ranges where subsequent peaks are expected, check which peaks occur in these intervals and try to find an optimal peak sequence. The interval ranges are defined relative to the last peak distance D, i.e. D is the distance between the last two peaks in propagation direction of the previously computed peak sequence. The continued peak sequence starts with the peak tuple for the previous frame and adds peaks to the left side if the peak propagation is to the left or to the right side, otherwise. Each new peak to be added is searched for in the

expected interval, while at the same time checking whether a similar peak exists in the interval that follows. Therefore, we have a peak look-ahead of 2. Each such peak pair – the two peaks looked ahead – is scored by computing their mean absolute height. The first peak of the pair that achieves the highest such score is installed as the definite next peak in the peak sequence. This peak propagation stops as soon as the center position of the addressed frame has passed by two peaks or if no further peak is found. It is deliberate that the score to evaluate peaks in unstable intervals considers only the absolute peak heights. A measure that accounted also for the peak distances would deliver false peak sequences, owing to the irregular peak distances that we expect in unstable intervals. Figure 9.4 illustrates the look-ahead strategy for successor peaks in left propagation direction, starting at peak p_0 that is part of peak tuple (p_0, p_1, p_R) of the previous frame. It shows the case of elongated periods V2. The interval where a first peak is expected is denoted by I_0. We find two possible candidate peaks in I_0, namely $p_{k(1)}$ and $p_{k(2)}$. For both candidate peaks, we look for possible look-ahead peaks. In Figure 9.4, the look-ahead peaks for $p_{k(2)}$ are shown, they are $p_{k(2,1)}$ and $p_{k(2,2)}$ in interval $I_{k(2)}$ that depends on the position of $p_{k(2)}$. The peak pair $p_{k(2)}$ and $p_{k(2,2)}$ achieves the highest score, and thus $p_{k(2)}$ is installed as the next valid peak in the peak propagation.

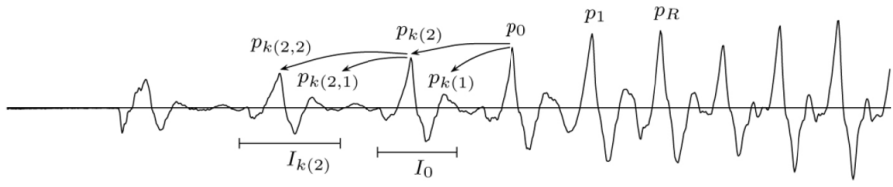


Figure 9.4. *Peak look-ahead strategy of 2 for the extended period case V2 that starts with tuple $P = (p_0, p_1, p_R)$ in propagation direction to the left. Peaks $p_{k(1)}$ and $p_{k(2)}$ are inspected from p_0 in interval I_0, and peaks $p_{k(2,1)}$ and $p_{k(2,2)}$ are the look-ahead peaks found in expected interval $I_{k(2)}$ starting from $p_{k(2)}$.*

The final step in the peak propagation stage is the evaluation of the various peak sequence variants. In general, the variant with the highest score, i.e. with the highest mean absolute peak height, is the best peak continuation sequence. However, some additional checks are needed to verify it. Here, we describe the evaluation procedure for the case where the previous frame has no octave jump. A similar procedure is applied if the previous frame is in an octave jump down state. In the case of currently no octave jump, we first check whether V1 and V2 deliver the same peak

sequence. If so, we keep V1 and discard V2. Otherwise, an additional peak propagation step for the next frame is performed to see whether V2 diverges in a double period case. If this is the case, V2 is discarded and V1 is kept. In all other cases, we keep the variant of either V1 or V2 with the larger score, i.e. the higher absolute mean peak height, in V1. Then, we evaluate V1 against the double period variant V3 if V3 has a score greater than or equal to V1. V3 is installed and the frame's octave jump state is set to "down" only if V3 has no middle peaks of sufficient heights, i.e. if the absolute height of a potential middle peak is smaller than a given percentage of the minimum of the absolute heights of the enclosing peaks. Otherwise, the peak tuple from the peak sequence of V1 is installed for the current frame.

9.5. Unstable voiced regions

Voiced regions without stable intervals or voiced regions that have no sufficiently large subsequences of equal F0 are treated in a separate post-processing step. Basically, the same propagation procedure is applied but the propagation starting or anchor point is found using looser conditions and additional information.

First, we compute the mean F0 of the last second of speech considering only frames with a verified F0, i.e. frames in equal sections of stable intervals used as starting points for F0 propagation or propagated frames in unstable intervals. We then compute candidate F0 values for all unstable frames of the voiced regions in the range of the mean F0. The permissible F0 range is the same as described in section 9.4.1. The anchor point for propagation is found by inspecting the equal section list of the stable intervals in the voiced regions, or a small section around the highest energy frame if no stable interval in the voiced region exists. The propagation starts from such a section if the mean of the F0 estimates does not deviate too largely from the last second's mean F0. If no such section takes place, we leave the F0 estimate unchanged. In this case, no propagation takes place.

9.6. Parallelization

The proposed F0 estimation algorithm can be parallelized in different ways. The algorithm delivers a segmentation of the audio signal into speech and pause segments that represent quasi-independent units. On the one hand, the whole F0 estimation algorithm may run on the speech segments in

parallel. On the other hand, the different tasks of the F0 estimation algorithm may run in parallel across the speech segments but sequentially within each speech segment. The three tasks of our algorithm are: (1) the preprocessing of computing the energies and the peaks, (2) the F0 estimation on stable intervals and (3) the peak propagation. A list of all signal peaks – both local minima and maxima – is maintained for each speech segment and computed at an early stage of processing. This second way of parallelized computation of F0 is implemented in our algorithm, as illustrated in Figure 9.5. The figure shows a series of speech and pause segments denoted by S and P. The three tasks (1)–(3) of the algorithm are depicted as blocks for each speech segment. These tasks are processed in different parallel processes T1, T2 and T3 across all speech segments. Of course, T2 has to wait until T1 has finished, and T3 waits until T2 has finished for the same speech segment.

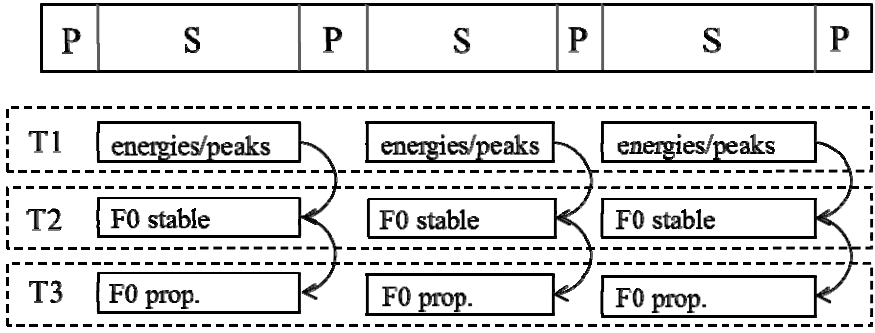


Figure 9.5. *Parallelized computation of F0 estimation. The three tasks are run in parallel across all speech segments but sequentially within each speech segment*

9.7. Experiments and results

The presented F0 estimation algorithm was evaluated on the Keele pitch reference database for clean speech. We measured the voiced error rate (VE), the unvoiced error rate (UE) and the gross pitch error rate (GPE). A voiced error is present if a voiced frame is recognized as unvoiced, an unvoiced error exists if an unvoiced frame is identified as voiced and a gross pitch error is counted if the estimated F0 differs by more than 20% from the reference pitch. The precision is given by the root-mean-square error (RMSE) in Hz for all frames classified as correct. The results for our parallel, cognition-oriented F0 estimation algorithm (PCO) are given in

Table 9.1. We also cite the results of other state-of-the-art F0 estimation or pitch detection algorithms where these figures were available: RAPT, PSHF and non-negative matrix factorization (NMF) [ROA 07, SHA 05, TAL 95]. RAPT is one of the best time-domain algorithms based on cross-correlation and dynamic programming.

	VE (%)	UE (%)	GPE (%)	RMSE (Hz)
PCO	*4.84*	*4.12*	*1.96*	*5.89*
RAPT	3.2	6.8	2.2	4.4
PSHF	4.51	5.06	0.61	2.46
NMF	7.7	4.6	0.9	4.3

Table 9.1. *Results of the parallel, cognition-oriented F0 estimation (PCO) in comparison with other state-of-the-art algorithms on the Keele pitch reference database*

The results show that the voiced and unvoiced error rates of PCO are comparable to those of the other state-of-the-art algorithms. In fact, the sum of both voiced and unvoiced error rates is the smallest for PCO, namely 8.96%, whereas it is 10% for RAPT, 9.57% for PSHF and 12.3% for NMF. The gross pitch error rate (GPE) of 1.96 is lower than that for RAPT but clearly not as low as that for the frequency-domain algorithms PSHF and NMF. Our algorithm is a pure time-domain method, and it performs better than RAPT which operates on the time domain also, but uses the normalized cross-correlation function. However, the gross pitch error (GPE) of our algorithm PCO is far lower than those of Praat, YIN and SAFE, as given in Table 9.2. The GPE of the cited algorithms are reported in [CHU 12]. For clarity, Table 9.2 also gives the GPE of the algorithms PSHF, NMF and RAPT, cited in Table 9.1.

	PSHF	NMF	*PCO*	RAPT	Praat	YIN	SAFE
GPE (%)	0.61	0.9	*1.96*	2.2	3.22	2.94	2.98

Table 9.2. *Gross pitch error rates alone of method PCO compared with standard F0 estimation algorithms on the Keele pitch reference database*

The root-mean-square error (RMSE), at 5.89 Hz – a measure for the preciseness of the correctly calculated F0 estimates – is higher than that for

the other algorithms, as given in Table 9.1. This can be explained as follows. The fundamental frequency F0 is defined as the inverse of the time between two minimum signal peaks or two maximum peaks. However, maximum or minimum peaks may have an inclination – either to the left or to the right – and often, there is a set of close peaks around the maximum or minimum peak, so that F0 is not as accurately calculated as with other methods. However, the accuracy of F0 estimates can certainly be improved by adjustment methods.

9.8. Conclusions

We have presented an F0 detection algorithm as an approximate model of the human cognitive process. It purely operates in the time domain, achieves very low error rates and outperforms the state-of-the-art correlation-based method RAPT in this respect. These results are achieved with little resources in terms of memory and computing power. Obviously, the strengths and the potential of the algorithm lie in the concepts that simulate human recognition of F0.

The question asked in the introduction whether F0 estimation using principles of human cognition can be performed equally well or better than the state-of-the-art F0 detection algorithms can be answered with a partial "yes" for clean speech. The gross pitch error rates of our algorithm are among the lowest of the standard F0 estimation algorithms. However, the accuracy in terms of root-mean-square error is still higher than that of other algorithms.

The presented algorithm is thoroughly extensible, as new special cases are easily implemented. In this sense, the algorithm can also be applied to other tasks, e.g. spontaneous or noisy speech, by analyzing the new cases and modeling them. In this way, it will become more and more generic. This procedure closely reflects human learning, which is said to function by adopting examples and building patterns independently of the frequency or probability of their occurrence [KUH 77]. For this reason, we have refrained from using weights or probabilities to favor one or other cases but instead to look ahead and evaluate until the case is decided.

A major strength of the algorithm is the segmentation of the speech signal into various structures additionally to the F0 contour. The recognition of

speech and pause segments makes a parallelization of the algorithm possible. The classification into stable and unstable intervals may be used for automatic speech recognition. Similarly to the presented F0 estimation, automatic speech recognition may first recognize the phonemes in stable intervals before detecting the phonemes in the unstable intervals. Spectra are more reliably computed on stable than on unstable intervals.

Future work will focus on extending the algorithm for both spontaneous and particularly noisy speech data and improving the accuracy of the F0 estimates.

9.9. Acknowledgments

The author wishes to thank Prof. Jozsef Szakos from the Hong Kong Polytechnic University for valuable comments and Prof. Guy Aston from the University of Bologna, Italy, for his careful proof-reading. She is also very grateful to Christian Singer who implemented the basic version of the pause-finding algorithm during his diploma thesis.

9.10. Bibliography

[ACH 05] ACHAN K., ROWEIS S., HERTZMANN A. *et al.*, "A segment-based probabilistic generative model of speech", *Proceedings of ICASSP*, pp. 221–224, 2005.

[BOE 01] BOERSMA P., "PRAAT, a System for doing phonetics by computer", *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[CHU 12] CHU W., ALWAN A., "SAFE: a statistical approach to F0 estimation under clean and noisy conditions", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 933–944, 2012.

[DEC 02] DE CHEVEIGNÉ A., KAWAHARA A., "YIN, a fundamental frequency estimator for speech and music", *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[EWE 10] EWENDER T., PFISTER B., "Accurate pitch marking for prosodic modification of speech segments", *Proceedings of INTERSPEECH*, pp. 178–181, 2010.

[GHA 14] GHAHREMANI P., BABA ALI B., POVEY D. *et al.*, "A pitch extraction algorithm tuned for automatic speech recognition", *Proceedings of INTERSPEECH*, pp. 2494–2498, 2014.

[GLA 15] GLAVITSCH U., HE L., DELLWO V., "Stable and unstable intervals as a basic segmentation procedure of the speech signal", *Proceedings of INTERSPEECH*, pp. 31–35, 2015.

[KAH 11] KAHNEMAN D., *Thinking, Fast and Slow*, Farrar, Straus and Giroux, New York, 2011.

[KUH 77] KUHN T.S., "Second thoughts on paradigms", *The Essential Tension, Selected Studies in Scientific Tradition and Change*, The University of Chicago Press, Chicago, pp. 837–840, 1977.

[MAR 72] MARKEL J.D., "The SIFT algorithm for fundamental frequency estimation", *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 5, pp. 367–377, 1972.

[MOO 08] MOORE B.C.J., *An Introduction to the Psychology of Hearing,* Emerald, Bingley, 2008.

[PEH 11] PEHARZ R., WOHLMAYR M., PERNKOPF F., "Gain-robust multi-pitch tracking using sparse nonnegative matrix factorization", *Proceedings of ICASSP*, pp. 5416–5419, 2011.

[PLA 95] PLANTE F., MEYER G.F., AINSWORTH W.A., "A pitch extraction reference database", *Proceedings of Eurospeech*, pp. 837–840, 1995.

[RAB 75] RABINER L.R., SAMBUR M.R., "An algorithm for detecting the endpoints of isolated utterances", *Bell System Technical Journal*, vol. 54, no. 2, 1975.

[RAB 76] RABINER L.R., CHENG M.J., ROSENBERG A.E. *et al*., "A comparative performance study of several pitch detection algorithms", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976.

[ROA 07] ROA S., BENNEWITZ M., BEHNKE S., "Fundamental frequency based on pitch-scaled harmonic filtering", *Proceedings of ICASSP*, pp. 397–400, 2007.

[ROD 11] RODERO E., "Intonation and emotion: influence of pitch levels and contour type on creating emotions", *Journal of Voice*, vol. 25, no. 1, pp. e25–e34, 2011.

[SEC 83] SECREST B.G., DODDINGTON G.R., "An integrated pitch tracking algorithm for speech systems", *Proceedings of ICASSP*, pp. 1352–1355, 1983.

[SHA 05] SHA F., SAUL L. K., "Real-time pitch determination of one or more voices by nonnegative matrix factorization", *Advances in Neural Information Processing Systems*, MIT Press, vol. 17, pp. 1233–1240, 2005.

[TAL 95] TALKIN D., *A Robust Algorithm for Pitch Tracking (RAPT)*, Speech Coding and Synthesis, Elsevier Science B.V., Amsterdam, 1995.

10

Benchmarking n-grams, Topic Models and Recurrent Neural Networks by Cloze Completions, EEGs and Eye Movements

Previous neurocognitive approaches to word predictability from sentence context in electroencephalographic (EEG) and eye movement (EM) data relied on cloze completion probability (CCP) data effortfully collected from up to 100 human participants. Here, we test whether three well-established language models can predict these data. Together with baseline predictors of word frequency and position in sentence, we found that the syntactic and short-range semantic processes of n-gram language models and recurrent neural networks (RNN) perform about equally well when directly accounting CCP, EEG and EM data. In contrast, a low amount of variance explained by a topic model suggests that there is no strong impact on the CCP and the N400 component of EEG data, at least in our Potsdam Sentence Corpus dataset. For the single-fixation durations of the EM data, however, topic models accounted for more variance, suggesting that long-range semantics may play a greater role in this earlier neurocognitive process. Though the language models were not significantly inferior to CCP in accounting for these EEG and EM data, CCP always provided a descriptive increase in explained variance for the three corpora we used. However, n-gram and RNN models can account for about half of the variance of the CCP-based predictability estimates, and the largest part of the variance that CCPs explain in EEG and EM data. Thus, our approaches may help to generalize neurocognitive models to all possible novel word combinations, and we propose to use the same benchmarks for language models as for models of visual word recognition.

Chapter written by Markus J. HOFMANN, Chris BIEMANN and Steffen REMUS.

10.1. Introduction

In neurocognitive psychology, manually collected cloze completion probabilities (CCPs) are the standard approach to quantifying a word's predictability from sentence context [KLI 04, KUT 84, REI 03]. Here, we test a series of language models in accounting for CCPs, as well as the data they typically account for, i.e. electroencephalographic (EEG) and eye movement (EM) data. With this, we hope to render time-consuming CCP procedures unnecessary. We test a statistical n-gram language model [KNE 95], a Latent Dirichlet Allocation (LDA) topic model [BLE 03], as well as a recurrent neural network (RNN) language model [BEN 03, ELM 90] for correlation with the neurocognitive data.

CCPs have been traditionally used to account for N400 responses as an EEG signature of a word's contextual integration into sentence context [DAM 06, KUT 84]. Moreover, they were used to quantify the concept of word predictability from sentence context in models of eye movement control [ENG 05, REI 03]. However, as CCPs are effortfully collected from samples of up to 100 participants [KLI 04], they provide a severe challenge to the ability of a model to be generalized across all novel stimuli [HOF 14], which also prevents their ubiquitous use in technical applications.

To quantify how well computational models of word recognition can account for human performance, Spieler and Balota [SPI 97] proposed that a model should explain variance at the item-level, i.e. latencies averaged across a number of participants. Therefore, a predictor variable is fitted to the mean word naming latency as a function of $y = f(x) = \sum a_n x_n + b + error$ for a number of n predictor variables x that are scaled by a slope factor a, an intercept of b, and an error term. The Pearson correlation coefficient r is calculated, and squared to determine the amount of explained variance r^2. Models with a larger number of n free parameters are more likely to (over-)fit error variance, and thus fewer free parameters are preferred (e.g. [HOF 14]).

While the best cognitive process models can account for 40–50% of variance in behavioral naming data [PER 10], neurocognitive data are noisier. The only interactive activation model that gives an amount of

explained variance in EEG data [BAR 07, MCC 81] was that of Hofmann *et al.* [HOF 08], who account for 12% of the N400 variance. Though models of eye movement control use item-level CCPs as predictor variables [ENG 05, REI 03], computational models of eye movement control have hardly been benchmarked at the item-level, to our knowledge [DAM 07].

While using CCP-data increases the comparability of many studies, the creation of such information is expensive and they only exist for a few languages [KLI 04, REI 03]. If it were possible to use (large) natural language corpora and derive the information leveraged from such resources automatically, this would considerably expedite the process of experimentation for under-resourced languages. Comparability would not be compromised when using standard corpora, such as that available through Goldhahn *et al*. [GOL 12] in many languages. However, it is not yet clear what kind of corpus is most appropriate for this enterprise, and whether there are differences in explaining human performance data.

10.2. Related work

Taylor [TAY 53] was the first to instruct participants to fill a cloze with an appropriate word. The percentage of participants who fill in the respective word serves as cloze completion probability. For instance, when exposed to the sentence fragment "He mailed the letter without a ___", 99% of the participants complete the cloze by "stamp", thus CCP equals 0.99 [BLO 80]. Kliegl *et al*. [KLI 04] logit-transformed CCPs to obtain $pred = ln(CCP/(1-CCP))$.

Event-related potentials are computed from human EEG data. For the case of the N400, words are often presented word-by-word, and the EEG waves are averaged across a number of participants relative to the event of word presentation. As brain-electric potentials are labeled by their polarity and latency, the term N400 refers to a negative deflection around 400 ms after the presentation of a target word.

After Kutas and Hillyard [KUT 84] discovered the sensitivity of the N400 to cloze completion probabilities, they suggested that it reflects the

semantic relationship between a word and the context in which it occurs. However, there are several other factors that determine the amplitude of the N400 [KUT 11]. For instance, Dambacher *et al*. [DAM 06] found that word frequency (*freq*), the position of a word in a sentence (*pos*), as well as predictability (*pred*) affect the N400.

While the eyes remain relatively still during fixations, readers make fitful eye movements called saccades [RAD 12]. When successfully recognizing a word in a stream of forward eye movements, no second saccade to or within the word is required. The time the eyes remain on that word is called single-fixation duration (SFD), which shows a strong correlation with word predictability from sentence context (e.g. [ENG 05]).

10.3. Methodology

10.3.1. *Human performance measures*

This study proposes that language models can be benchmarked by item-level performance on three datasets that are openly available in online databases. Predictability was taken from the Potsdam Sentence Corpus[1], first published by Kliegl *et al*. [KLI 04]. The 144 sentences consist of 1,138 tokens, available in Appendix A of [DAM 09], and the logit-transformed CCP measures of word predictability were retrieved from Ralf Engbert's homepage1 [ENG 05]. For instance, in the sentence "Manchmal sagen Opfer vor Gericht nicht die volle Wahrheit" [Before the court, victims tell not always the truth.], the last word has a CCP of 1. N400 amplitudes were taken from the 343 open-class words published in Dambacher and Kliegl [DAM 07]. These are available from the Potsdam Mind Research Repository[2]. The EEG data published there are based on a previous study (see [DAM 06] for method details). The voltage of 10 centroparietal electrodes was averaged across up to 48 artifact-free participants from 300 to 500 ms after word presentation for quantifying the N400. SFD are based on the same 343 words from Dambacher and Kliegl [DAM 07], available from the same source URL. Data were included when this word was only fixated for

1 http://mbd.unipotsdam.de/EngbertLab/Software.html
2 http://read.psych.unipotsdam.de

one time, and these SFDs ranged from 50 to 750 ms. The SFD was averaged across up to 125 German native speakers [DAM 07].

10.3.2. *Three flavors of language models*

Language models are based on a probabilistic description of language phenomena. Probabilities are used to pick the most fluent of several alternatives, e.g. in machine translation or speech recognition. Word **n-gram models** are defined by a Markov chain of order $n-1$, where the probability of the following word only depends on previous $n-1$ words. In statistical models, the probability distribution of the vocabulary, given a history of $n-1$ words, is estimated based on n-gram counts from (large) natural language corpora. There exist a range of n-gram language models (see, e.g., Chapter 3 in [MAN 99], which are differentiated by the way they handle unseen events and perform probability smoothing). Here, we use a Kneser–Ney [KNE 95] 5-gram model[3]. For each word in the sequence, the language model computes a probability p in]0; 1[. We use the logarithm *log(p)* of this probability as a predictor. We used all words in their full form, i.e. did not filter for specific word classes and did not perform lemmatization. N-gram language models are known to model local syntactic structure very well. Since only n-gram models use the most recent history for predicting the next token, they fail to account for long-range phenomena and semantic coherence (see [BIE 12]).

Latent Dirichlet Allocation (LDA) topic models [BLE 03] are generative probabilistic models representing documents as a mixture of a fixed number of N topics, which are defined as unigram probability distributions over the vocabulary. Through a sampling process like Gibbs sampling, topic distributions are inferred. Words frequently co-occurring in the same documents receive a high probability in the same topics. When sampling the topic distribution for a sequence of text, each word is randomly assigned to a topic according to the document-topic distribution and the topic-word distribution. We use Phan and Nguyen's [PHA 07] GibbsLDA implementation for training an LDA model with 200 topics (default values for $\alpha = 0.25$ and $\beta = 0.001$) on a background corpus. Words occurring in too many documents (a.k.a. stopwords) or too few documents (mistyped or rare words) were removed from the LDA vocabulary. Then, retain the per

3 https://code.google.com/p/berkeleylm/

document topic distribution $p(z|d)$ and the per topic word distribution $p(w|z)$, where z is the latent variable representing the topic, d refers to a full document during training – during testing d refers to the history of the current sentence – and w is a word. In contrast to our earlier approach using only the top three topics [BIE 15], we here computed the probability of the current word w given its history d as a mixture of its topical components $p(w|d) = p(w|z)p(z|d)$. We hypothesize that topic models account for some long-range semantic aspects missing in n-gram models. While Bayesian topic models are probably the most widespread approach to semantics in psychology (e.g. [GRI 07]), latent semantic analysis (LSA) is not applicable in our setting [LAN 97]: we use the capability of LDA to account for yet unseen documents, whereas LSA assumes a fixed vocabulary and it is not trivial to fold new documents into LSA's fixed document space.

While Jeff Elman's [ELM 90] seminal work suggested early on that semantic and also syntactic structure automatically emerges from a set of simple recurrent units, such an approach has received little attention in language modeling for a long time, but is currently of interest to many computational studies. In brief, such **Neural Network Language Models** are based on the optimization probability of the occurrence of a word, given its history using neural units linking back to themselves, much as the neurons in the CA3 region of the human hippocampus [MAR 71, NOR 03]. The task of language modeling using neural networks was first introduced by Bengio *et al*. [BEN 03] and received at that point only little attention because of computational challenges regarding space and time complexity. Due to recent advancement in the field of neural networks – for an overview, see [MIK 12] – neural language models gained more popularity, particularly because of the so-called neural word embeddings as a side product. The language model implementation we use in this work is a recurrent neural network architecture[4] similar to the one used by Mikolov's Word2Vec[5] toolkit [MIK 13]. We trained a model with 400 hidden layers and hierarchical softmax. For testing, we used the complete history of a sentence up to the current word.

4 FasterRNN: https://github.com/yandex/faster-rnnlm
5 Word2Vec: https://code.google.com/archive/p/word2vec/

10.4. Experiment setup

Engbert *et al.*'s [ENG 05] data are organized in 144 short German sentences with an average length of 7.9 tokens, and provide features, such as *freq* as corpus frequency in occurrences per million [BAA 95], *pos* and *pred*. We test whether two corpus-based predictors can account for predictability, and compare the capability of both approaches in accounting for EEG and EM data. For training n-gram and topic models, we used three different corpora differing in size and covering different aspects of language. Further, the units for computing topic models differ in size.

NEWS: a large corpus of German online newswire from 2009, as collected by LCC [GOL 12], of 3.4 million documents/30 million sentences/ 540 million tokens. This corpus is not balanced, i.e. important events in the news are covered better than other themes. The topic model was trained on the document level.

WIKI: a recent German Wikipedia dump of 114,000 articles/7.7 million sentences/180 million tokens. This corpus is rather balanced, as concepts or entities are described in a single article each, independent of their popularity, and spans all sorts of topics. The topic model was trained on the article level.

SUB: German subtitles from a recent dump of opensubtitles.org, containing 7,420 movies/7.3 million utterances/54 million tokens. While this corpus is much smaller than the others, it is closer to a colloquial use of language. Brysbaert *et al.* [BRY 11] showed that word frequency measures of subtitles provide numerically greater correlations with word recognition speed than larger corpora of written language. The topic model was trained on the movie level.

Pearson's product-moment correlation coefficient was calculated (e.g. [COO 10, p. 293]), and squared for the N = 1,138 predictability scores [ENG 05] or N = 343 N400 amplitudes or SFD [DAM 07]. To address overfitting, we randomly split the material into two halves, and test how much variance can be reproducibly predicted on two subsets of 569 items. For N400 amplitude and SFD, we used the full set, because one half was too small for reproducible predictions. The correlations between all predictor

variables can be examined in Table 10.1. We observe very high correlations between the n-gram and the RNN predictions within and across corpora. The correlations involving topic-based predictions are smaller, supporting our hypothesis that they reflect a somewhat different neurocognitive process.

		1.	2.	3.	4.	5.	6.	7.	8.	9.
NEWS	1. n-gram		0.65	0.87	0.87	0.56	0.84	0.83	0.59	0.80
	2. topic	0.65		0.68	0.66	0.78	0.70	0.61	0.77	0.61
	3. neural	0.87	0.68		0.84	0.59	0.88	0.77	0.62	0.79
WIKI	4. n-gram	0.87	0.66	0.84		0.61	0.90	0.79	0.59	0.78
	5. topic	0.56	0.78	0.59	0.61		0.65	0.55	0.75	0.55
	6. neural	0.84	0.70	0.88	0.90	0.65		0.76	0.64	0.79
SUB	7. n-gram	0.83	0.61	0.77	0.79	0.55	0.76		0.61	0.85
	8. topic	0.59	0.77	0.62	0.59	0.75	0.64	0.61		0.61
	9. neural	0.80	0.61	0.79	0.78	0.55	0.79	0.85	0.61	

Table 10.1. *Correlations between the language model predictors*

10.5. Results

10.5.1. *Predictability results*

In the first series of results, we examine the prediction of manually obtained CCP-derived predictability with corpus-based methods. A large amount of explained variance would indicate that predictability could be replaced by automatic methods. As a set of baseline predictors, we use *pos* and *freq*, which explains 0.243/0.288 of the variance for the first and the second half of the dataset, respectively. We report results in Table 10.2 for all single corpus-based predictors alone and in combination with the baseline, all combinations of the baseline with n-gram, topics and neural models from the same corpus.

Predictors	NEWS	WIKI	SUB
n-gram	0.262/0.294	0.226/0.253	0.268/0.272
topic	0.063/0.061	0.042/0.040	0.040/0.034
neural	0.229/0.226	0.211/0.226	0.255/0.219
base+n-gram	**0.462/0.490**	0.423/0.458	0.448/0.459
base+topic	0.348/0.375	0.333/0.357	0.325/0.355
base+neural	**0.434/0.441**	0.418/0.433	**0.447/0.418**
base+n-gram+topic	0.462/0.493	0.427/0.464	0.447/0.458
base+n-gram+neural	0.466/0.492	0.431/0.461	0.467/0.461
base+neural+topic	0.438/0.445	0.421/0.436	0.446/0.423
base+n-gram+topic+neural	0.466/0.493	0.433/0.465	0.467/0.460

Table 10.2. *r^2 explained variance of predictability, given for two halves of the dataset, for various combinations of baseline and corpus-based predictors*

It is apparent that the n-gram scores best, and also the neural model alone reaches r^2 levels that approach the baseline. In contrast, much as our earlier top-three topics approach [BIE 15], the mixture of all topics explains only a relatively low amount of variance. Combining the baseline with the n-gram predictor already reaches a level very close to the combination of all predictors, thus it may provide the best compromise between parsimony and explained variance. Again, this model performance is closely followed by the recurrent neural network (see Figure 10.1).

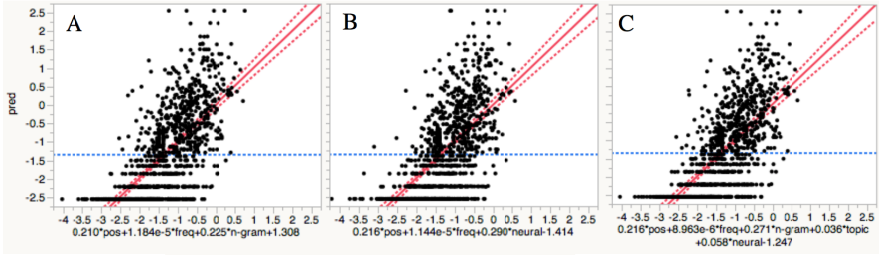


Figure 10.1. *Prediction models exemplified for the NEWS corpus in the x-axes and the N = 1,138 predictability scores on the y-axes. A) Prediction by baseline + n-gram (r^2 = 0.475), B) a recurrent neural network (r^2 = 0.437) and C) a model containing all predictors (r^2 = 0.478). The three pairwise Fisher's r-to-z tests revealed no significant differences in explained variance (Ps > 0.18). For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

We also fitted a model based on all corpus-based predictors from all corpora, which achieved the overall highest r^2 (0.490/0.507). In summary, it becomes clear that about half of the empirical predictability variance can be explained by a combination of positional and frequency features combined with either a word n-gram language model, or a recurrent neural network.

10.5.2. *N400 amplitude results*

For modeling N400, we have even more combinations at our disposal, since we can combine corpus-based measures with the baseline, the predictability performance and with both. We evaluate on all 343 data points for N400 amplitude fitting. Without using corpus-based predictors, the baseline predicts a mere 0.032 of variance, predictability alone explains 0.192 of variance and their combination explains 0.193 of variance – i.e. the baseline is almost entirely subsumed by CCP-based predictability. As can be observed from Table 10.3, this is a score that is not yet reached by the language models, even when combining all of them.

Predictors	NEWS	WIKI	SUB
n-gram	0.141	0.140	0.126
topic	0.039	**0.055**	0.025
neural	0.108	0.098	0.100
base+n-gram	**0.161**	0.153	0.135
base+topic	0.063	0.079	0.055
base+neural	0.133	0.116	0.114
base+n-gram+topic	0.161	0.158	0.132
base+n-gram+neural	0.167	0.153	0.141
base+neural+topic	0.133	0.123	0.112
base+n-gram+topic+neural	**0.167**	0.158	0.137
base+n-gram+pred	**0.223**	0.226	0.206
base+topic+pred	0.193	0.204	0.191
base+neural+pred	0.221	0.212	0.206
base+n-gram+topic+pred	0.225	0.228	0.203
base+n-gram+neural+pred	0.228	0.226	0.209
base+neural+topic+pred	0.224	0.215	0.203
base+n-gram+topic+neural+pred	**0.232**	0.228	0.206

Table 10.3. *r^2 explained variance of the N400 for various combinations of the corpus-based predictors, in combination with the baseline, and with the empirical predictability*

When comparing the performance of the computationally defined predictors, a picture similar to the prediction of the empirical predictability emerges. The n-gram model scores best, particularly for the larger NEWS and WIKI corpora. This confirms a generally accepted hypothesis that larger training data trumps smaller, more focused training data, see e.g. [BAN 01] and others. The n-gram model is, however, immediately followed by the neural model, and again, the topic predictor provides the poorest performance in explaining N400 amplitude variance, which suggests that the N400 does not reflect long-range semantic processes. The best combination without predictability, with a score of $r^2 = 0.167$, approaches the performance of the predictability and baseline (see Figure 10.2).

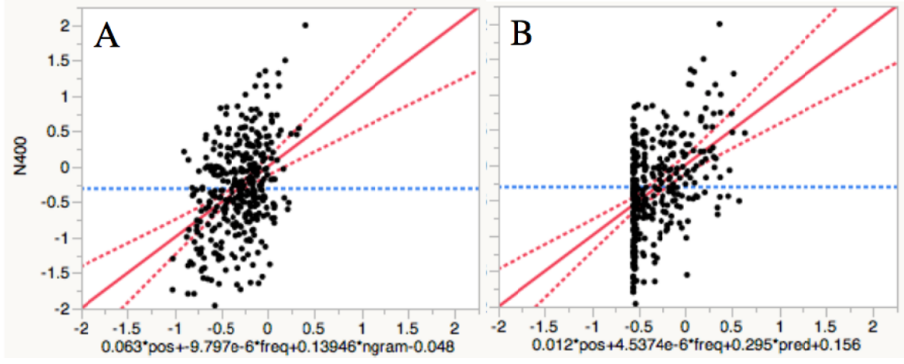


Figure 10.2. *Prediction models exemplified for the NEWS corpus in the x-axes and the N = 334 mean N400 amplitudes on the y-axes. A) Prediction by baseline + n-gram ($r^2 = 0.161$), and B) a standard approach to N400 data, consisting of the baseline of position and frequency, as well as the empirical predictability ($r^2 = 0.193$; e.g. [DAM 06]). Fisher's r-to-z tests revealed no significant differences in explained variance (P = 0.55). For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

The experiments with predictability as an additional predictor confirm the results from the previous section: n-grams + baseline and predictability capture slightly different aspects of human reading performance, thus their combination explains up to 6% more net variance than predictability alone.

10.5.3. *Single-fixation duration (SFD) results*

Finally, we examine the corpus-based predictors for modeling the mean single fixations duration for 343 words. For this target, the *pos+freq* baseline explains $r^2 = 0.021$, whereas predictability, alone or combined with the baseline, explains $r^2 = 0.184$.

Predictors	NEWS	WIKI	SUB
n-gram	0.225	0.140	0.126
topic	0.135	0.140	0.100
neural	0.242	0.190	**0.272**
base+n-gram	0.239	0.226	0.226
base+topic	0.152	0.154	0.127
base+neural	0.265	0.204	**0.284**
base+n-gram+topic	0.260	0.262	0.246
base+n-gram+neural	0.287	0.238	0.297
base+neural+topic	0.279	0.235	0.298
base+n-gram+topic+neural	0.295	0.265	**0.307**
base+n-gram+pred	0.273	0.274	0.258
base+topic+pred	0.235	0.250	0.229
base+neural+pred	0.314	0.267	0.320
base+n-gram+topic+pred	0.297	0.301	0.275
base+n-gram+neural+pred	0.319	0.283	0.322
base+neural+topic+pred	0.319	0.289	0.329
base+n-gram+topic+neural+pred	0.323	0.304	**0.330**

Table. 10.4. *Explained variance of the single-fixation durations, for various combinations of baseline, predictability and corpus-based predictors*

The experiments confirm the utility of n-gram models in accounting for eye movement data. The n-gram model alone explains even more variance than predictability – however, the difference is not significant ($P > 0.46$).

In contrast to the previous approaches to predictability and N400 amplitudes, however, the recurrent neural network outperformed the n-gram

model at a descriptive level, as it accounted for up to 3% more of the variance than the n-gram model. This performance was not reached at the largest NEWS corpus, but at the smaller SUB corpus. This suggests that – for SFD data – the dimension reduction seems to compensate for the larger amount of the noise in the smaller training dataset (see [BUL 07, GAM 16, HOF 14]). Therefore, the neural model may provide a better fit for such early neurocognitive processes when it is trained by colloquial language [BRY 11].

The topics model seems to have a stronger impact on SFDs than on the other neurocognitive benchmark variables, suggesting a greater influence of long-range semantics on SFDs than on predictability or the N400. Taken together, these findings suggest that SFDs reflect different cognitive processes than the N400 (see [DAM 07]).

Last but not least, though again adding predictability increased the total amount of explained variance by 2%, the language models did an excellent job in accounting for SFD data. When taking all language model-based predictors together, this accounts for significantly more variance than the standard model using predictability (see Figure 10.3).

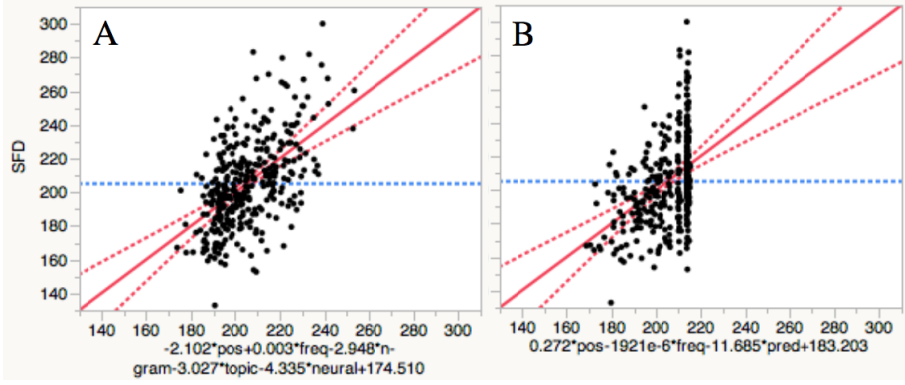


Figure 10.3. *Prediction models exemplified for the SUB corpus in the x-axes and the N = 334 mean SFD scores on the y-axes. A) Prediction by baseline + all three language models (r^2 = 0.295), and B) a standard approach to SFD data, using the baseline and predictability as predictors of SFDs (r^2 = 0.184). Fisher's r-to-z test revealed a significant difference in explained variance (z = 1.95; P = 0.05). For a color version of this figure, see www.iste.co.uk/sharp/cognitive.zip*

10.6. Discussion and conclusion

We have examined the utility of three corpus-based predictors to account for word predictability from sentence context, as well as the EEG signals and EM-based reading performance elicited by it. Our hypothesis was that word n-gram models, topic models and recurrent neural network models would account for the predictability of a token, given the preceding tokens in the sentence, as perceived by humans, as well as some electroencephalographic and eye movement data that are typically explained by it. Therefore, we used the amount of explained item-level variance as a benchmark, which has been established as a standard evaluation criterion for neurocognitive models of visual word recognition (e.g. [HOF 08, PER 10, SPI 97]).

Our hypothesis was at least partially confirmed: n-gram models and RNNs, sometimes in combination with a frequency-based and positional baseline, are highly correlated with human predictability scores and in fact explain variance of human reading performance to an extent comparable to predictability – slightly less on the N400 but slightly more on the SFD. This, however, might at least in part be explainable by a larger amount of noise in the EEG data with fewer participants when compared with the eye movement data with much more participants.

The long-range semantic relationships as captured by topic models, on the other hand, provided a different picture. If any, the topic model made only a minor contribution to predictability and the N400. For the fast and successful recognition of a word at one glance, as reflected by SFDs in contrast, long-range document level relationships seem to provide a stronger contribution. This result pattern occurs even in the context of single sentences, without a discourse level setting the topic of a document. This suggests that the colloquial and taxonomic far-reaching semantic long-term structure particularly determines the fast and effective single-glance recognition of a word within the first 300 ms after the onset of word recognition. In contrast, topic models hardly account for somewhat later processes around 400 ms in the brain-electric data and the time-consuming, probably late, processes being contained in the predictability scores.

For predicting the empirical word predictability from sentence context as well as the N400, recurrent neural network models often performed somewhat worse than the n-gram approach. For predicting SFDs, however, the neural model was superior. Most interestingly, the neural network model

performs best when it is trained on a small but probably more representative sample of everyday language. Therefore, size probably does not trump everything and in any model [BAN 01]. It also hints at the generalization properties of its dimensionality reduction, which are more important for smaller training data [BUL 07, GAM 16, HOF 14], but probably leads to imprecise modeling when more data are available.

Can we now safely replace human predictability scores with n-gram statistics? Given the high correlation between predictability and the combination of n-grams with frequency and positional information, and given that n-gram-based predictors achieve similar levels of explained variance to predictability, the answer seems to be positive. However, though our corpus-based approaches explain most of the variances that manually collected CCP scores also account for, adding predictability always accounts for more variance – though this difference is not significant (see Figure 10.2; cf. Figures in [BIE 15]).

When contrasting the standard predictors of position, frequency and predictability used in eye tracking and EEG research (e.g. [DAM 06, REI 03]), only for the SFDs, all three corpus-based predictors did a better job than the standard model. However, with this approach, it is apparent that many more predictors are needed, and thus the probability for fitting error variance is much larger than that for the standard model. Thus, we think that much more evidence is required, before we dare to state this as a firm conclusion. Also for this three-predictor model, adding the empirical predictability provides a net gain of 2% explained variance.

As n-gram or neural models together with word frequency and position captured about half of the predictability variance, and most of the N400 and SFD variance elicited by it, we propose that it can be used to replace tediously collected CCPs. This not only saves a lot of pre-experimental work, but also opens the possibility to apply (neuro-) cognitive models in technical applications. For instance, n-gram models, topic models and neural models can be used to generalize computational models of eye movement control to novel sentences [ENG 05, REI 03].

In the end, language models can also improve our understanding of the cognitive processes underlying predictability, EEG and EM measures. While it is not clear what exactly determines human CCP-based predictability performance, the different language models provide differential grain size

levels using their training data, thus paving the way for the question as to which neurocognitive measures of "word predictability" are affected by sentence- or document-level semantic knowledge. While Ziegler and Goswami [ZIE 05] discussed the optimal grain size of language learning at the word-level and sub-word-level grain sizes, recent evidence of a severe decline of comprehension abilities since the 1960s suggests the necessity to continue with that discussion at the level of supralexical semantic integration [SPI 16].

10.7. Acknowledgments

The "Deutsche Forschungsgemeinschaft" (MJH; HO 5139/2-1), the German Institute for Educational Research in the Knowledge Discovery in Scientific Literature (SR) program and the LOEWE Center for Digital Humanities (CB) supported this work.

10.8. Bibliography

[BAA 95] BAAYEN H.R., PIEPENBROCK R., GULIKERS L., The CELEX Lexical Database. Release 2 (CD-ROM), LDC, University of Pennsylvania, Philadelphia, 1995.

[BAN 01] BANKO M., BRILL E., "Scaling to very very large corpora for natural language disambiguation", *Proceedings of ACL '01*, Toulouse, pp. 26–33, 2001.

[BAR 07] BARBER H.A., KUTAS M., "Interplay between computational models and cognitive electrophysiology in visual word recognition", *Brain Research Reviews*, vol. 53, no. 1, pp. 98–123, 2007.

[BEN 03] BENGIO Y., DUCHARME R., VINCENT P. *et al.*, "A neural probabilistic language model", *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.

[BIE 12] BIEMANN C., ROOS S., WEIHE K., "Quantifying semantics using complex network analysis", *Proceedings of COLING 2012*, Mumbai, pp. 263–278, 2012.

[BIE 15] BIEMANN C., REMUS S., HOFMANN M.J., "Predicting word 'predictability' in cloze completion, electroencephalographic and eye movement data", *Proceedings of the 12th International Workshop on Natural Language Processing and Cognitive Science*, Krakow, pp. 83–94, 2015.

[BLE 03] BLEI D.M., NG A.Y., JORDAN M.I. "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[BLO 80] BLOOM P.A., FISCHLER I., "Completion norms for 329 sentence contexts", *Memory & cognition*, vol. 8, no. 6, pp. 631–642, 1980.

[BUL 07] BULLINARIA J.A., LEVY J.P., "Extracting semantic representations from word co-occurrence statistics: a computational study", *Behavior Research Methods*, vol. 39, no. 3, pp. 510–526, 2007.

[BRY 11] BRYSBAERT M., BUCHMEIER M., CONRAD M. *et al.*, "A review of recent developments and implications for the choice of frequency estimates in German", *Experimental psychology*, vol. 58, pp. 412–424, 2011.

[COO 10] COOLICAN H., *Research Methods and Statistics in Psychology*, Hodder & Stoughton, London, 2010.

[DAM 06] DAMBACHER M., KLIEGL R., HOFMANN M.J. *et al.*, "Frequency and predictability effects on event-related potentials during reading", *Brain Research*, vol. 1084, no. 1, pp. 89–103, 2006.

[DAM 07] DAMBACHER M., KLIEGL R., "Synchronizing timelines: relations between fixation durations and N400 amplitudes during sentence reading", *Brain research*, vol. 1155, pp. 147–162, 2007.

[DAM 09] DAMBACHER M., *Bottom-up and Top-down Processes in Reading*, Potsdam University Press, Potsdam, 2009.

[ELM 90] ELMAN J.L., "Finding structure in time," *Cognitive Science*, vol. 211, pp. 1–28, 1990.

[ENG 05] ENGBERT R., NUTHMANN A., RICHTER E.M. *et al.*, "SWIFT: a dynamical model of saccade generation during reading", *Psychological Review*, vol. 112, no. 4, pp. 777–813, 2005.

[GAM 16] GAMALLO P., "Comparing explicit and predictive distributional semantic models endowed with syntactic contexts," *Language Resources and Evaluation*, pp. 1–17, doi:10.1007/s10579-016-9357-4, 2016.

[GOL 12] GOLDHAHN D., ECKART T., QUASTHOFF U., "Building large monolingual dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages", *Proceedings of LREC 2012*, Istanbul, pp. 759–765, 2012.

[GRI 07] GRIFFITHS T.L., STEYVERS M., TENENBAUM J.B., "Topics in semantic representation", *Psychological Review*, vol. 114, no. 2, pp. 211–244, 2007.

[HOF 14] HOFMANN M.J., JACOBS A.M., "Interactive activation and competition models and semantic context: from behavioral to brain data", *Neuroscience & Biobehavioral Reviews*, vol 46, pp. 85–104, 2014.

[HOF 08] HOFMANN M.J., TAMM S., BRAUN M.M. *et al.*, "Conflict monitoring engages the mediofrontal cortex during nonword processing", *Neuroreport*, vol. 19, no. 1, pp. 25–29, 2008.

[KLI 04] KLIEGL R., GRABNER E., ROLFS M. *et al.*, "Length, frequency, and predictability effects of words on eye movements in reading", *European Journal of Cognitive Psychology*, vol. 16, no. 12, pp. 262–284, 2004.

[KNE 95] KNESER R., NEY H., "Improved backing-off for m-gram language modeling", *Proceedings of IEEE Int'l Conference on Acoustics, Speech and Signal Processing*, Detroit, pp. 181–184, 1995.

[KUT 11] KUTAS M., FEDERMEIER K.D., "Thirty years and counting: finding meaning in the N400 component of the event-related brain potential (ERP)", *Annual Review of Psychology*, vol. 62, pp. 621–647, 2011.

[KUT 84] KUTAS M., HILLYARD S.A., "Brain potentials during reading reflect word expectancy and semantic association", *Nature*, vol. 307, no. 5947, pp. 161–163, 1984.

[LAN 97] LANDAUER T.K., DUMAIS S.T., "A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge", *Psychological Review*, vol. 104, no. 2, pp. 211–240, 1997.

[MAN 99] MANNING C.D., SCHÜTZE H., *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, 1999.

[MAR 71] MARR D., "Simple memory: a theory", *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 262, no. 841, pp. 23–81, 1971.

[MCC 81] MCCLELLAND J.L., RUMELHART D.E., "An interactive activation model of context effects in letter perception: part 1", *Psychological Review*, vol. 5, pp. 375–407, 1981.

[MIK 12] MIKOLOV T., Statistical language models based on neural networks, PhD Thesis, Brno University of Technology, 2012.

[MIK 13] MIKOLOV T., YIH W., ZWEIG G., "Linguistic regularities in continuous space word representations", *Proceedings of NAACL-HLT*, Atlanta, pp. 746–751, 2013.

[NOR 03] NORMAN K.A., O'REILLY R.C., "Modeling hippocampal and neocortical contributions to recognition memory: a complementary-learning-systems approach", *Psychological Review*, vol. 110, no. 4, pp. 611–646, 2003.

[PER 10] PERRY C., ZIEGLER J.C., ZORZI M., "Beyond single syllables: large-scale modeling of reading aloud with the Connectionist Dual Process (CDP++) model", *Cognitive Psychology*, vol. 61, no. 2, pp. 106–151, 2010.

[PHA 07] PHAN X.-H., NGUYEN C.-T., "GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation (LDA)", available at: http://gibbslda.sourceforge.net/, 2007.

[RAD 12] RADACH R., GÜNTHER T., HUESTEGGE L., "Blickbewegungen beim Lesen, Leseentwicklung und Legasthenie", *Lernen und Lernstoerungen*, vol. 1, no. 3, pp. 185–204, 2012.

[REI 03] REICHLE E.D., RAYNER K., POLLATSEK A., "The E-Z reader model of eye-movement control in reading: comparisons to other models", *The Behavioral and Brain Sciences*, vol. 26, no. 4, pp. 445–476, 2003.

[SPI 16] SPICHTIG A., HIEBERT H., VORSTIUS C. *et al.*, "The decline of comprehension-based silent reading efficiency in the U.S.: a comparison of current data with performance in 1960", *Reading Research Quarterly*, vol. 51, no. 2, pp. 239–259, 2016.

[SPI 97] SPIELER D.H., BALOTA D.A., "Bringing computational models of word naming down to the item level", *Psychological Science*, vol. 8, no. 6, pp. 411–416, 1997.

[TAY 53] TAYLOR W.L., "'Cloze' procedure: a new tool for measuring readability", *Journalism Quarterly*, vol. 30, p. 415, 1953.

[ZIE 05] ZIEGLER J.C., GOSWAMI U., "Reading acquisition, developmental dyslexia, and skilled reading across languages : a psycholinguistic grain size theory", *Psychological Bulletin*, vol. 131, no. 1, pp. 3–29, 2005.

This page intentionally left blank

List of Authors

Chris BIEMANN
University of Hamburg
Germany

Philippe BLACHE
Brain and Language Research Institute
CNRS–University of Provence
Aix-en-Provence
France

Alexandre BLONDIN-MASSÉ
University of Quebec at Montreal
Canada

Mohamed Amine BOUKHALED
Paris VI University
France

Jean-Gabriel GANASCIA
Paris VI University
France

Izabela GATKOWSKA
Jagiellonian University
Kraków
Poland

Debela Tesfaye GEMECHU
Addis Ababa University
Ethiopia

Ulrike GLAVITSCH
Swiss Federal Laboratories for Materials Science and Technology (EMPA)
Zurich
Switzerland

Maciej GODNY
Jagiellonian University
Kraków
Poland

Stevan HARNAD
University of Quebec at Montreal
Canada
and
University of Southampton
UK

Markus J. HOFMANN
University of Wuppertal
Germany

Michał KORZYCKI
AGH
University of Technology
Kraków
Poland

Marcos LOPES
University of São Paolo
Brazil

Mélanie LORD
University of Quebec at Montreal
Canada

Wiesław LUBASZEWSKI
Jagiellonian University
Kraków
Poland

Odile MARCOTTE
University of Quebec at Montreal
Canada

Marcello PELILLO
European Centre for Living
Technology (ECLT)
Ca' Foscari University
Venice
Italy

Reinhard RAPP
University of Mainz
and
University of Applied Sciences
Magdeburg-Stendal
Germany

Steffen REMUS
University of Hamburg
Germany

Florence SÈDES
Paul Sabatier University
Toulouse
France

Bernadette SHARP
Staffordshire University
Stoke-On-Trent
England

Rocco TRIPODI
European Centre for Living
Technology (ECLT)
Ca' Foscari University
Venice
Italy

Philippe VINCENT-LAMARRE
Ottawa University
Canada

Michael ZOCK
Traitement Automatique du
Langage Ecrit et Parlé (TALEP)
Laboratoire d'Informatique
Fondamentale (LIF)
CNRS
Marseille
France

Index

S, T, V, W

As natural language processing spans many different disciplines, it is sometimes difficult to understand the contributions and the challenges that each of them presents. This book explores the special relationship between natural language processing and cognitive science, and the contribution of computer science to these two fields. It is based on the recent research papers submitted at the international workshops of Natural Language and Cognitive Science (NLPCS) which was launched in 2004 in an effort to bring together natural language researchers, computer scientists, and cognitive and linguistic scientists to collaborate together and advance research in natural language processing.

The chapters cover areas related to language understanding, language generation, word association, word sense disambiguation, word predictability, text production and authorship attribution. This book will be relevant to students and researchers interested in the interdisciplinary nature of language processing.

Bernadette Sharp is Professor of Applied Artificial Intelligence (AI) at Staffordshire University, UK. Her research interests include AI, natural language processing, and text mining. She has been Chair and Editor of the International Workshop for Natural Language Processing and Cognitive Science since 2004.

Florence Sèdes is Professor of Computer Science at Toulouse University, France. Her research areas cover information systems and data management with applications dedicated to multimedia, metadata and mobility in ambient intelligence, social media and CCTV. She supervises a "smart restaurant" platform for emotion and social interaction analysis, and contributes to the ISO 22311 standard.

Wiesław Lubaszewski is Professor at the Department of Computational Linguistics of the Jagiellonian University and Professor at the Computer Science Department of AGH, University of Technology, in Kraków, Poland. His research interests include natural language dictionaries, text understanding, knowledge representation, and information extraction.

ELSEVIER

ISTE PRESS

www.iste.co.uk

9 781785 482533