TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRONIC AND COMPUTER ENGINEERING
INTELLIGENT SYSTEMS DIVISION

Semantic Composition in DSMs: Activational Priming and Transformational Properties for Similarity Modeling

by

Spiros Georgiladakis

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE MASTER OF SCIENCE OF

ELECTRONIC AND COMPUTER ENGINEERING

December 18, 2015

THESIS COMMITTEE

Professor Euripides Petrakis, *Thesis Supervisor*
Associate Professor Alexandros Potamianos
Associate Professor Polychronis Koutsakis

Abstract

Distributional Semantic Models (DSMs) have been successful at modeling the meaning of words in isolation. Interest has recently shifted to compositional structures, i.e., lexical units that comprise of words that represent individual concepts, such as phrases and sentences. Network DSMs (NDSMs) represent and handle semantics via operations on word neighborhoods, i.e., semantic graphs comprising of a target lexical unit's semantically most similar words. Semantic networks are based on activational priming, a cognitively-based theory that a specific area which shares common features can be activated upon the triggering of a related stimulus. In this thesis, a variety of activation composition and similarity modeling strategies is proposed that aims to address compositionality within the framework of the respective layers of NDSMs. In the activation layer, we propose several activation schemes, motivated by psycholinguistics, that utilize variable size activations in order to compose neighborhoods for complex structures. In the similarity layer, we model similarity metrics that operate on the derived neighborhoods to estimate similarity. The proposed schemes cover a range of approaches for modeling semantics in complex structures. We also investigate modifier properties and transformational models from the literature, and propose a fusion scheme that regulates the transformational properties of phrase modifiers in order to weight the contribution of its component models for handling semantics. To this end, the model utilizes network and transformational models under a fusion scheme that models similarity. It is shown that, by fusing strictly compositional with transformational models to realise a flexible model that adapts to phrase behavior by considering modifier properties, performance gains can be achieved.

Περίληψη

Τα Κατανεμημένα Σημασιολογικά Μοντέλα (ΚΣΜ) έχουν καταστεί επιτυχή όσον αφορά τη μοντελοποίηση νοήματος για απομονωμένες λέξεις. Το ενδιαφέρον έχει πρόσφατα μετακινηθεί σε συνθετικές δομές, δηλαδή, σε λεκτικές μονάδες που συντίθενται από λέξεις που εκπροσωπούν διακριτές έννοιες, όπως φράσεις και προτάσεις. Τα Δικτυακά ΚΣΜ (ΔΚΣΜ) εκπροσωπούν και χειρίζονται σημασιολογική πληροφορία μέσα από επενέργειες σε γειτονιές λέξεων, δηλαδή, σημασιολογικούς γράφους που αποτελούνται από τις ομοιότερες σημασιολογικά λέξεις σε σχέση με την εν λόγω λεκτική μονάδα. Τα σημασιολογικά δίκτυα βασίζονται στην ενεργοποιητική προέγερση, μία θεωρία βασισμένη στη γνωσιακή επιστήμη κατά την οποία μία συγκεκριμένη περιοχή που μοιράζεται κοινά χαρακτηριστικά δύναται να ενεργοποιηθεί υπό το έναυσμα κάποιου σχετικού ερεθίσματος. Στην εργασία αυτή προτείνονται μια ποικιλία από στρατηγικές σύνθεσης ενεργοποιητικών περιοχών και μοντελοποίησης ομοιότητας με σκοπό την αντιμετώπιση της διαδικασίας σύνθεσης στο πλαίσιο των σχετικών επιπέδων στα ΔΚΣΜ. Στο επίπεδο ενεργοποίησης, προτείνουμε διάφορα σχέδια ενεργοποίησης, παρακινούμενα από θεωρίες ψυχογλωσσολογίας, που χρησιμοποιούν ενεργοποιητικές περιοχές μεταβλητού μεγέθους με στόχο τη σύνθεση γειτονιών για σύνθετες δομές. Στο επίπεδο ομοιότητας, μοντελοποιούμε μετρικές ομοιότητας που λειτουργούν στις προκύπτουσες γειτονιές για τον υπολογισμό ομοιότητας. Τα προτεινόμενα σχήματα καλύπτουν ένα εύρος προσεγγίσεων για τη μοντελοποίηση σημασιολογικής πληροφορίας σε σύνθετες δομές. Επιπλέον, διερευνούμε τις ιδιότητες λέξεων που επέχουν ρόλο τροποποιητή, καθώς και μετασχηματικά μοντέλα από τη βιβλιογραφία, και προτείνουμε μία στρατηγική σύντηξης που χρησιμοποιεί τις μετασχηματικές ιδιότητες τροποποιητών σε φράσεις με σκοπό την στάθμιση της συμβολής των επιμέρους μοντέλων για το χειρισμό της σημασιολογικής πληροφορίας. Για το σκοπό αυτό, το μοντέλο χρησιμοποιεί μοντέλα βασισμένα σε δίκτυα και σε μετασχηματικές στρατηγικές κάτω από ένα συγχωνευτικό σχήμα, με στόχο τη μοντελοποίηση ομοιότητας. Αποδεικνύεται ότι, συντήξει αυστηρά συνθετικών και μετασχηματικών μοντέλων για την υλοποίηση ενός ευέλικτου μοντέλου που προσαρμόζεται στη συμπεριφορά των φράσεων και στις ιδιότητες των τροποποιητών, μπορούν να επιτευχθούν οφέλη στην απόδοση.

Acknowledgements

I would like to express my sincere gratitude to Prof. Euripides Petrakis for offering me his support whenever I needed it. His trust in me was crucial for me to be able to follow the course of my graduate studies. Also, I'd like to express my gratitude and appreciation to Prof. Alexandros Potamianos for his continuous guidance throughout my research efforts and through the writing of the current thesis. Without his invaluable assistance, suggestions and expertise, it wouldn't be possible for me to reach the level of knowledge I acquired. Special thanks to all the members of the committee for their valuable comments and suggestions during the review of this thesis.

During the course of my research I had the full support and assistance of Dr. Elias Iosif, who was always available to provide me with new suggestions and ideas. Our constructive discussions and his generous willingness to help proved essential for overcoming the inevitable difficulties that emerged during my research.

I would like to acknowledge the funding I received from the PortDial ("Language Resources for Portable Multilingual Spoken Dialog Systems") and SpeDial ("Machine-Aided Methods for Spoken Dialogue System Enhancement and Customization for Call-Center Applications") projects, supported by the EU Seventh Framework Programme (FP7), grant numbers, grant numbers 296170 and 611396 respectively, and the invaluable experience I gained from participating in them.

Last but not least, I would like to express my gratitude to my parents for supporting me during these years and my appreciation to all my friends and Aliki that helped me when the load seemed high. Special thanks to Anais for his quiet support during the writing of the thesis, as well as to Ludovico Einaudi, whose music helped me stay focused and concentrated on task.

Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing
NLU	Natural Language Understanding
VSM	Vector Space Model
DS	Distributional Semantics
DSM	Distributional Semantic Model
IR	Information Retrieval
IC	Information Content
NDSM	Network Distributional Semantic Model
MSE	Mean Squared Error
PPMI	Positive Pointwise Mutual Information
SVD	Singular Value Decomposition
NMF	Non-negative Matrix Factorization
LSR	Least Squares Regression
RR	Ridge Regression

Chapter 1

Introduction

"An isolated word, or a detail of a design, can be understood. But the meaning of the whole escapes. Once we know the number one, we believe that we know the number two, because one plus one equals two. We forget that first we must know the meaning of plus."

- Alpha 60 Supercomputer, Alphaville

1.1 Background

Natural language is a communication tool that has evolved over the course of many millenia, for the purpose of efficient transmission of ideas and in order to promote understanding. Humans have subconsciously established an ability to use natural language to their advantage by intuitively relating expressions, concepts and meanings, with specific symbols, words and phrases, within a specific context. Concepts that relate to either abstract or concrete meanings can be mapped to these language segments, that can in turn be representative of these meanings. *Natural Language Processing* (NLP) is the field of computer science that is related to the area of human - computer interaction, in terms of *Natural Language Understanding* (NLU). NLP challenges include the comprehension of natural language by computers, i.e., providing them with the ability to deduct concepts and, subsequently, *meaning*, by using natural language as input. *Semantics* is called the study of this meaning. Computational semantics study the automation of processing the development of and reasoning with representations of meaning by natural language expressions. Meaning can be defined as the concept that is represented by a word, phrase, etc. Concepts are initially composed of sensory experience, and further enriched by our reflections upon our subjective sensory observations. When we use words and symbols to reference those concepts, we create language. Broadly, the current thesis deals with the ability of computers to utilize language lexicalizations, and the manner in which they interact, in order to derive the respective meanings that they reference. The conceptual-

ization of meanings can be subsequently used to approximate the semantic proximity, i.e., semantic relatedness, between different language lexicalizations. Semantics serve as vital components for numerous other natural language applications as well, such as word sense disambiguation, machine translation, semantic role labeling, information extraction, paraphrasing and textual entailment [1], grammar induction, question answering, semantic taxonomy, sentiment and affective text analysis [2], etc.

The process of estimating semantics from single-word tokens (unigrams) has reached a mature state, and much progress has been made with respect to measuring word-level semantic similarity. In general, word similarities cannot rely on simple lexical distance metrics. The words "word" and "lord", for example, however close they are in terms of lexical distance, infer a completely different meaning. This cannot be captured by comparing their lexicalizations alone. Current approaches to address the problem of *semantic similarity* can be generally divided into three main categories:

1. those that rely on a *knowledge base* and its structure, in order to cluster words into predefined categories and semantic concepts (i.e., classes), subsequently estimating similarity between two words as a function of the distance between their classes,

2. those that instead follow a *data-driven* approach, using raw data (e.g., a unannotated corpus) and statistics extracted them in order to derive semantics and similarities, and

3. hybrid approaches that attempt to utilize advantages from both directions by combining both strategies.

In this thesis, we will focus on data-driven approaches for computing semantics and for the estimation of semantic similarity via the use of statistical methods, applied on raw input data. Concerning data-driven approaches, estimating word semantics is typically realised by exploiting the distributional hypothesis of meaning, i.e., the hypothesis that semantically close linguistic lexicalizations tend to share similar linguistic environments [3, 4]. To this end, word semantics can be estimated by extracting statistics from the context in which they occur, and can then be used to represent the words they refer. This is typically realised by encoding those semantics into *vectors*. Vectors are generally populated by information regarding the co-occurrence of a word with another, typically within a specified window size (i.e., context). Various techniques are then used to normalize and encode this information into high semantic spaces, such as feature reweighing,

dimensionality reduction techniques, etc. These high-dimensional spaces, used for concept representation, are known as *Vector Space Models* (VSMs) [5]. VSMs act as the building block for *Distributional Semantics* (DS) and *Distributional Semantic Models* (DSMs) [6]. DSMs adopt the geometric metaphor of meaning: information is presented as coordinates in a geometry space, according to which words are projected as points in the space [7]. Word co-occurrence statistics are, thus, considered as features, used to populate respective feature vectors or, more generally, feature tensors [6, 8]. Those feature tuples can then be utilized for various semantic tasks and applications. Their similarity, for example, can be computed as the proximity between the corresponding points in the geometry space, i.e., similarity between words can be estimated by performing simple algebraic operations on VSMs.

VSMs have proved to be efficient at capturing word semantics, and, within the framework of DSMs, have been successful at estimating word semantic similarity [5]. However, their application for semantic representation of longer and more complex structures, as is the case with phrases or sentences, is, in itself, a more complex task. Meaning from such structures derives as the result of various compositional phenomena, which are inherent properties of natural language creativity [9]. Moreover, language creativity itself creates limitations regarding the adaptation of methods, developed to estimate word semantics, to the case of more complex structures. These complex structures are composed by words that infer coherent meanings in isolation, and are essential components of the derived meaning of the structure. This has motivated approaches regarding *compositional semantics*. Compositional semantics adopt the idea that the meaning of a complex linguistic expression, i.e., a structure that is composed by constituent words that represent meanings of their own, derives by the meanings of its parts, and can be modified by the manner in which these parts relate and interact. The composed meaning is, thus, derived by also considering the function of relations among the constituents, which is not always trivial to detect and represent in computational semantics. The key idea behind recent approaches in semantic composition using DSMs is the representation of words as vectors and the composition of meaning through their combination, using simple composition schemes, e.g., vector addition or multiplication [10, 11], or via other proposed functions that consider additional word properties and linguistic phenomena. Regardless of the function used, the composed representations adhere to the paradigm of VSMs, while the cosine between the resulting vectors is used for estimating semantic similarity. Such efforts have proved to be effective when computing similarity between simple bigram structures, however, their limitations

were revealed for the case of longer structures [12], where the composition of meaning becomes more complex.

1.2 Motivation and Problem Statement

Interest in the research community of NLP has been shifted from words to larger and more complex linguistic structures, such as phrases, sentences, or complete documents, for the task of measuring semantic similarity. Typically, word-level semantics are estimated by considering their co-occurrence within a common context, while their semantic similarity can be computed by directly comparing their feature vectors. However, regarding bigrams, or larger and more complex linguistic structures in general, such approaches are bounded by unavoidable restrictions in data availability. The effectiveness of this approach is also inversely proportional to the length and rareness of the structure in the data, while inherent properties of language creativity also reveal the limitations of such approaches, when considering large linguistic segments. In particular, for every word that is added to a linguistic expression, the composition's frequency of occurrence within a given corpus decreases. Moreover, linguistic expressions that span over an area of text, cannot be easily detected within a corpus. For example, consider the bigrams "black car" and "quiet neighborhood" within the sentence "he saw a seemingly black, although not quite sure of the color, car, drive through the quiet and abandoned neighborhood". In this example, the formation of said bigrams is intercepted by other words or linguistic segments of the sentence, rendering the detection of the bigrams a difficult task. Based on the aforementioned, compositional semantics are motivated by the information that resides within the structure, and are concerned with the way that the meaning of the structure is composed both from the meaning of its constituent words, and from the additional meaning that is extracted from the relation among those constituents, i.e., the meaning that is encoded within the functional relations within the structure itself. If word semantics can be estimated, comparing two phrases by estimating the similarities between the words that constitute them could serve as a solution to the problem. Nevertheless, intrinsic information by itself can only partially address the problem, since the meaning of multi-word phrases encodes information that is not always a direct product of the composition of their parts. Another problem derives from variations of word senses, which may also depend on their use within a given context, such as the use of "back" as the back of a person or as an adverb. Last but not least, phrases can be used in different ways, based on their literal

of figurative applicability within a context. For instance, the phrase "piece of cake" can be used whether to refer to something that is trivial to accomplish, or to an actual piece of cake. Therefore, the selected approach for estimating semantics greatly depends on the information and the fashion that this information is encoded in the target phrases, i.e., through their constituents and the relations among them. The problem is, thus, whether and how appropriate semantics for a phrase can be derived from the semantics of its components, without losing information during this process that is essential for tasks, such as that of semantic similarity. This has become an important issue in distributional semantics [10, 13–15], as it is not yet clear which way of combining the components is best suited for which tasks, and many attempts to address this problem tend to result in ad-hoc realised systems. Research has been primarily focused on measuring the similarity between simple structures. These structures are mainly bigram lexicalizations, such as noun-noun, or adjective-noun constructions. Recently, attention has been shifted towards longer phrases, or even complete sentences.

1.3 Contribution and Organization

In this thesis, an alternative approach is proposed for estimating similarity between phrases, based on the notion of *semantic networks*, that serve as an added layer built on top of DSMs. To this end, a recent network-based implementation of DSMs [16] has been extended in [17], in order to handle semantics of compositional structures. The used framework consists of activation models motivated by semantic priming [18]. It is proposed that each structure activates a specific area that is regarded as a sub-space residing within the semantic network (i.e., a semantic neighborhood). The novelty of the present work is two-fold:

1. First, strategies targeted on different perspectives are proposed for composing activation areas for compositional structures, within a framework alternative to VSMs.

2. Second, we utilize transformational properties of compositions to determine the contribution of different approaches to similarity estimation under a fused approach.

In addition, the role of words as operators on the meaning of the structures they occur in is studied, by measuring their transformative degree. The contribution of this thesis resides on the proposal of various approaches that consider different linguistic properties of a structure, for the computation of semantic neighborhoods. It also proposes a novel idea

that considers the transformational degree of a given phrase, in order to estimate similarity as a fusion of different compositional models.

The remainder of this thesis is organized as follows: in Chapter 2, various models of semantic similarity from the literature are described, while the methods that accompany such models are presented. In Chapter 3, the notion of NDSMs is presented, along with the literature and the proposed approaches, regarding their utilization for semantic composition and similarity estimation between structures. The fusion model, integrating NDSMs with transformational models from the literature, is also described in this chapter, while the trasformative properties of modifiers is investigated. In Chapter 4, we describe the experimental procedure that was followed for evaluating the models, and discuss on the findings. The thesis is concluded in Chapter 5, with model applications on the sentential level, conclusions and future work.

Chapter 2

Models of Semantic Similarity

In this chapter, semantic models from the literature that estimate semantic relatedness or similarity are separated and presented into two types of categories. The first category, referred to as knowledge-based models, consists of models that estimate similarities with respect to explicitly defined knowledge about the world, structured into lexical databases. The second category, referred to as distributional semantic models, considers similarity as a statistically inferred function of context word distributions. It should be pointed out that semantic relatedness does not imply semantic similarity, even though the inverse is valid. Semantic relatedness metrics utilize all possible relations between words in order to conclude about the degree in which they are related, while semantic similarity metrics merely make use of hierarchical types of relations, such as hyponymy or hyperymy.

2.1 Knowledge-based Models

The knowledge-based approach in semantic estimation is focused in methods that, in addition to linguistic information, also depend on explicitly defined domain or world knowledge. This knowledge can aid models in solving problems such as ambiguity resolution, or inferencing [19]. Knowledge is explicitly formulated in lexical networks, such as Roget's Thesaurus [20, 21], the WordNet lexical database [22], or, more recently, Wikipedia. These resources encode relational dependencies that exist between words, such as synonymy or antonymy, hypernymy or hyponymy, entailment, meronymy, etc. A drawback of handmade lexical structures lies in their size, as they provide limited coverage. They also have scalability issues, since creating such resources requires lexicographic expertise, as well as a lot of cost in time and effort. Furthermore, these resources have strong lexical orientation and tend to consider word semantics in isolation, rather than generic world knowledge [23]. The majority of knowledge-based systems is domain specific, which makes addressing of specific tasks easier, since much of the ambiguity that is present in terms, in the generic knowledge, can be eliminated by focusing on a specific domain. However, that is not always the case, since some applications may have to interpret input from multiple domains.

A large number of metrics has been defined, that computes semantic relatedness or similarity using various properties of the underlying graph structure of those resources [24–30]. In this section, a brief presentation of the most widely-used knowledge based metrics is displayed, based on their qualitative performance in various language processing applications. A knowledge database is typically represented as a hierarchically structured lexical network that is composed by nodes that each defines a specific concept, i.e., a cluster of semantically related words, and edges, that refer to a type of relation between two connected concepts. To this end, let c denote a concept, and i and j be two words that belong to concepts c_i and c_j, respectively. Also, let $l(c_i, c_j)$ define the shortest path between c_i and c_j, i.e., the minimum length between the two concepts, computed in terms of nodes or edges. Also, let $d(c_i)$ refer to the depth of c_i, defined as the length of the path from the hierarchy root r to c_i, i.e., $d(c_i) = l(r, c_i)$ (similarly for $d(c_j)$).

Lesk similarity. The Lesk similarity is defined as an overlap function between definitions of i and j, and is based on an algorithm that was proposed in [31] as a solution for word sense disambiguation.

Hirst and St-Onge's Relatedness. Hirst and St-Onge [32] proposed that semantic relatedness between c_i and c_j is inversely proportional to the size of $l(c_i, c_j)$, as well as to the number of times $t(c_i, c_j)$ that the direction changes in WordNet, when we move from c_i to c_j. The path from c_i to c_j could follow three directions: a) horizontal: c_j is an antonym of c_i, b) upward: c_j is a hypernym or meronym of c_i, or c) downward: c_j is a hyponym or holonym of c_i. To this end, the authors proposed a relatedness measure that estimates the strength of the relationship by

$$R_{HS}(c_i, c_j) = \alpha - l(c_i, c_j) - \beta t(c_i, c_j), \tag{2.1}$$

where α and β serve as weighting parameters.

Sussna's Depth-relative Scaling. A drawback of Hirst and St-Ogne's Relatedess measure is the implicit assumption that edges are equally weighted, when considering semantic relatedness. Sussna tried to address this issue by considering a range of weights, for each relation r [33, 34]. The core idea was to normalize the weight of each edge for each relation r that originates from c_i, $q(c_i, r)$, by the total number of edges of the same type that also originate from the same node, $e_r(c_i)$, as

$$q(c_i, r) = \max_r - \frac{\max_r - \min_r}{e_r(c_i)}. \tag{2.2}$$

Then, the semantic distance between c_i and c_j, $D_s(c_i, c_j)$, is defined as the sum of their respective weights across the directions that originate from both concepts, r and r', normalized by the maximum concept depth.

$$D_S(c_i, c_j) = \frac{q(c_i, r) + q(c_j, r')}{2 \max\{d(c_i), d(c_j)\}} \tag{2.3}$$

This approach was motivated by the hypothesis that concepts positioned low in the hierarchical structure of the network tend to be more similar than those positioned at the upper levels.

Wu & Palmer's Conceptual Similarity. This scaled metric, proposed in [35], combines the depths of c_i and c_j in the WordNet taxonomy, and the depth of their lowest common subsumer (LCS), into a similarity score as

$$S_{WP}(c_i, c_j) = \frac{2d(LCS(c_i, c_j))}{d(c_i) + d(c_j)} \tag{2.4}$$

Leacock & Chodorow's Normalized Path Length. In Leacock & Chodorow [36], $l(c_i, c_j)$ is normalized by the maximum length D in the taxonomy and a similarity metric is defined as

$$S_{LC}(c_i, c_j) = -log \frac{l(c_i, c_j)}{2D}, \tag{2.5}$$

where l is computed via node counts. 2.5 is defined as a similarity metric as it merely considers *IsA* relations in the WordNet taxonomy.

Resnik's Information-based approach. Resnik proposed that similar concepts tend to share similar information. The measure introduced by Resnik [37] returns the Information Content (IC) of $LCS(c_i, c_j)$. IC is defined as

$$IC(c) = -logp(c), \tag{2.6}$$

$p(c)$ being the probability of occurrence of an instance of concept c in a corpus. To this end, Resnik's measure is defined as

$$S_{Res}(c_i, c_j) = IC(LCS(c_i, c_j)). \tag{2.7}$$

Lin's Universal Similarity. Lin [28] attempted to derive a measure that can be used both universally, and at the same time be theoretically valid. By using the IC as a measure of commonality between two concepts, Lin enhanced Resnik's similarity measure by adding

a normalization factor, comprising of the IC of c_i and c_j as

$$S_{Lin}(c_i, c_j) = \frac{2IC(LCS(c_i, c_j))}{IC(c_i) + IC(c_j)}, \tag{2.8}$$

where IC is defined by Eq. 2.7.

Jiang & Conrath's Combined approach. Resnik's approach restricts the role of network edges in estimating semantic proximity, as they are solely utilized to detect the superordinate of a concept pair. Limitations arise when attempting to distinguish between different pairs of concepts that however share the same superordinate: in those cases, an edge-based method might be more appropriate. In order to eliminate the drawbacks of Resnik's approach, Jiang and Conrath [29] attempted to merge both edge- and node-based techniques by using network edges for similarity computations, and post-correct them by adapting IC to the form of conditional probabilities. In particular, they proposed that semantic distance between child concept c_i and parent concept c_k (i.e., $c_k = par(c_i)$) is correlated with the conditional probability $p(c_i|c_k)$, i.e., the probability of encountering an instance of c_i, given an instance of its parent concept c_k, such as that

$$D_{JC}(c_i, c_k) = -logp(c_i|c_k). \tag{2.9}$$

By adopting Resnik's framework for mapping concepts with probabilities, and based on Eq. 2.6, Eq. 2.9 becomes

$$D_{JC}(c_i, c_k) = IC(c_i) - IC(c_k). \tag{2.10}$$

2.10 can be adapted to estimate similarity between an arbitrary pair of concepts c_i and c_j, within a taxonomy, via Jiang and Conrath's similarity metric, defined as

$$S_{JC}(c_i, c_j) = \frac{1}{IC(c_i) + IC(c_j) - 2IC(LCS(c_i, c_j))}. \tag{2.11}$$

Recently, using Wikipedia for computing semantic relatedness has attracted interest in the respective field and has been investigated as an alternative knowledge base for estimating relatedness or similarity [38]. In spite of its short lifespan (2001), Wikipedia provides entries on a vast number of entities and even specialized concepts. Moreover, entity relations, defined by the way in which Wikipedia articles are interlinked, can be interpreted in different ways according to the task at hand, as they encode implicit semantic relations that are not present among WordNet concepts and relations, that has typically been used as

the standard knowledge database. In [39], semantic relatedness is investigated in the spectrum of three measure categories, namely, path-based, IC-based, and text-overlap-based measures. Wikipedia has been found to perform equally, if not better, than WordNet. In [23], they use machine learning and propose a method called Explicit Semantic Analysis (ESA) in order to represent text meanings in high-dimensional spaces. These spaces are categorized as concepts, derived from Wikipedia. Apart from the described drawbacks, knowledge-based solutions have provided a viable alternative to statistically or grammar-based approaches that solely depend on linguistic information [25, 40].

2.2 Distributional Semantic Models

DSMs are typically constructed by co-occurrence statistics of word tuples. These statistics are extracted from a raw input source, such as a corpus, and various techniques are applied in order to post-process them into high-dimensional semantic spaces. The procedure for constructing DSMs is typically based on the following steps. First, co-occurrence counts are extracted from text. This can be realised via two different approaches, that also categorize DSMs into structured and unstructured. Structured DSMs parametrize syntactic relationships between words, which are utilized for surface analysis and extraction of attributes. To this end, co-occurrence statistics are computed as corpus-derived triples, which are typically composed by pairs of words and the syntactic or lexico-syntactic relations between them. The approach is motivated by the assumption that these relations encode semantic properties that are shared between these words [41–44]. Unstructured DSMs instead adopt a bag-of-words model, i.e., they regard text as an unordered multi-set of its words, without consideration of grammatical or syntactical properties [3, 6]. They, thus, consider all context as feature properties of a target word i, utilizing co-occurrence counts to represent distributed information. For instance, given the sentence *"A boy plays football with its friends in a huge park"*, the words *"boy"* and *"park"* share a degree of semantic features, according to unstructured DSMs, since they co-occur under the same context, i.e., the words *"a", "plays", "football", "with", "its", "friends", "in", "a", "huge"* [1]. A feature vector is therefore created that represents i as $\vec{i} = (t_{i,1}, ..., t_{i,k}, ..., t_{i,n})$, where $t_{i,k} \geq 0$ and n is equal to the vocabulary size. $t_{i,k}$ is computed considering all occurrences of i in the corpus. This is realised using a binary scheme that assigns 1 to $t_{i,k}$, if t_k occurs within the $2H + 1$ context window to the left or to the right of i, otherwise $t_{i,k}$ is assigned

[1] In the case of the context window size being equal to sentence length.

0. This binary scheme is typically re-weighted, and the dimensionality of the matrix that the set of vectors comprise is reduced. In the rest of the section, the standard techniques upon which DSMs are built are described.

2.2.1 Feature Weighting

Co-occurrence statistics constitute the raw vectorial representation of words. In their simplest form, these VSMs are composed by binary vectors, i.e., a vectorial tuple that indicates whether a (target) word occurs with a (context) word within a specified size. Most frequently, vectorial features are instead composed of co-occurrence counts, i.e., the frequencies in which words occur within the same window size with each other. In the majority of the cases, these features need to be normalized. Feature normalization can be implemented according to various schemes inspired from Information Retrieval (IR), such as tf-idf, log-likelihood, etc.

2.2.2 Dimensionality Reduction

Co-occurrence matrices, in their original size, tend to be greatly large and sparse, while, in their most part, they are composed of zero-valued elements. To this end, some type of dimensionality reduction technique typically follows the computation of vectors, in order to reduce the size of the matrix and to transcend the encoded semantics into higher and more qualitative dimensional spaces. The most-widely used technique for dimensionality reduction is **Singular Value Decomposition (SVD)**. SVD is based on linear algebra and has its origins in IR, and was proposed in [45]. Let X be the semantic space that a VSM comprises, i.e., a word-context matrix. The key idea behind SVD is the factorization of matrix X by considering three low-dimensional matrices, U, Σ, and V, so that their product can form a low-rank approximation of X. Specifically, SVD computes the decomposition

$$X \approx U_k \Sigma_k V_k^T, \tag{2.12}$$

where k is the desired dimension rank, and returns $U_k\Sigma_k$, truncated to the $min(k, rank(X))$ dimension. This approximation can assist in a significant reduction of the dimensions of X, typically from tens of thousands to just a few hundred. SVD is the most typically selected technique in order to create semantic spaces that address the task of semantic similarity [46]. Another dimensionality reduction technique used is the **Non-negative Matrix Factorization (NMF)**, proposed by [47]. NMF factorizes X into an $n \times r$

matrix W and an $r \times k$ matrix H. In particular, NMF computes the factorization of X into two non-negative matrix factors [2], W and H, such as that

$$X \approx WH. \tag{2.13}$$

The W and H matrices are selected in order to minimize $\| X - WH \|_2$. Typically, r is chosen to be smaller that n or k, which results in H being a compressed version of the original matrix to the reduced dimension k. In other words, each vector $x \in X$ can be approximated by a linear combination of the columns of W, weighted by the values of $h \in H$ such as that $x \approx Wh$. Thus, W acts as an optimized function that serves the role of transforming h into x.

2.2.3 Regression Techniques

Many models employ regression in order to learn the appropriate functions by empirical fitting, rather than being set their parameters by manual configuration. This is useful for composing a supervised model that can be adapted and used for various tasks and applications, without the need for human intervention for setting the required parameters. Learning is realised by training the model on input data, which is is typically done in DSMs using regression techniques that try to learn the weights that best approximate observed examples. A standard approach in regression analysis, subsequently inherited for training language models, is by using **(Partial) Least Squares Regression (LSR)**. Given an input matrix X, derived after being passed through a dimensionality reduction technique, such as SVD, LSR finds the matrix weights that solve

$$X = arg\ min(\| AX - B \|_2). \tag{2.14}$$

Another regression technique, proposed in [48], is **Ridge Regression (RR)**, also known as L_2 regularized regression. When multicollinearity is present, the matrix $X^T X$ becomes almost singular, while the diagonal elements of $(X^T X)^{-1}$ become quite large, along with weight variance. RR attempts to handle the problem of multicollinearity using a different approach than LSR, specifically by employing the use of a (positive) constant λ in order to strengthen the non-singularity of the matrix $X^T X$ by adding λ to its diagonal elements. λ can be set using a predefined value, or it can be tuned using generalized cross-validation [49]. The objective function is then learnt by minimizing least square error using provided

[2] X should also be a non-negative matrix for applying NMF

input examples, i.e., it is trained to find the matrix which solves

$$X = arg\ min(\| AX - B \|_2 + \lambda \| X \|_2). \tag{2.15}$$

As a result, the learnt weight matrix B produces competitive results at a higher speed [50].

2.2.4 Word Embeddings: Context-predicting DSMs

Traditionally, contextual information has been the standard for providing semantic representations to word meanings. This standard is based on the distributional hypothesis of meaning, and the assumption that semantically similar words have a higher tendency to share similar contextual distributions [3, 4]. DSMs are, by definition, derived by vectors that provide information about word co-occurrence within specific context. These vectors are in turn re-weighted and transformed, in order to achieve better compression and in order to improve performance, in an indirectly unsupervised way. However, the process of selecting the best configurations for each step involves decision making and can lead to ad-hoc models that, in many cases, fail to perform at the same level, when applied in different semantic tasks. Also, the problem of language creativity becomes an apparent obstacle, when trying to compute representations for all possible combinations of words. During the last few years, a new set of models has emerged within the field of DSMs. In these models, which base their definitions on deep learning [51], the process is reversed and the weights in a word vector are instead set in order to maximize the probability of the contexts they occur in. Vector weights are, thus, learnt, in order to best approximate, or *predict*, the representation of their respective word's contextual features [52, 53]. This approach eventually leads to the same desired result: similar vectors will be shared among words with similar meaning, since their weights will be trained using similar context. These models, referred to as *context-predicting DSMs*, or *word embeddings*, have attracted the interest of the field, since they use the same resources as traditional DSMs, such as unannotated corpora, while they replace the need of specifying step-by-step parameters with a single, well-defined training step [54]. These language models also address the problem of language creativity, since they approximate word vectors, instead of relying on definite word counts in order to compute them. To this end, the models appear to constantly gain support in comparison with the traditional DSMs.

2.2.5 Measures of Semantic Similarity

As a first step regarding data-driven approaches, as are DSMs, contextual features are extracted from an input source, such as a corpus, within the basis of the distributional hypothesis of meaning. These features are then regarded as semantic properties of the words they represent. Feature extraction is typically realised by measuring the strength of relations between the words of a given vocabulary. To this end, various methods have been proposed that estimate the degree of such relations. Since the information that is represented in DSMs encodes semantic properties of words, those methods comprise metrics of semantic similarity between those words. Those metrics can be separated into two categories, specifically a) co-occurrence-, and b) context-based.

Co-occurrence metrics

Co-occurrence-based are motivated by the assumption that semantic relatedness between two words is associated with their co-existence within the same corpus unit D. The corpus unit can be of any level, e.g., a document, a paragraph or a sentence. Let $|D|$ define the total number of corpus units, $|D|i|$ being the set of units that are being indexed by word i.

Jaccard and Dice coefficient. The Jaccard coefficient measure calculates the semantic diversity between sets and is defined as

$$S_{Jacc}(i,j) = \frac{|D|i,j|}{|D|i| + |D|j| - |D|i,j|}. \tag{2.16}$$

The measure estimates the maximum likelihood of the probability of a document where i and j co-occur, over the probability of a document where either i or j occurs. It terms of corpus unit sets, it is defined as the size of their intersection, divided by the size of their union. The Dice coefficient measure is based on 2.16 and is computed as

$$S_{Dice}(i,j) = \frac{2|D|i,j|}{|D|i| + |D|j|}. \tag{2.17}$$

Normalized Google Distance. Proposed in [55, 56] and motivated by Kolmogorov complexity. Normalized Google Distance (G) is defined as

$$D_G(i,j) = \frac{\max\{log|D|i|, log|D|j|\} - log|D|i,j|}{log|D| - \min\{log|D|i|, log|D|j|\}}. \tag{2.18}$$

Google Semantic Relatedness. In [57], a variation of D_G is proposed as a similarity

measure that is bounded within the $[0, 1]$ range. To this end, Google Semantic Relatedness (S_G) is defined as

$$S_G(i, j) = e^{-2D_G(i,j)}, \tag{2.19}$$

where $D_G(i, j)$ is computed as in Eq. 2.18.

Mutual Information. A typically used method for measuring co-occurrence statistics is by computing the Mutual Information (MI) between two words. MI is computed by comparing the probability of observing the words together, with the probabilities of observing them in isolation, as in

$$MI(i, j) = log_2 \frac{p(i, j)}{p(i)p(j)} \tag{2.20}$$

The measure captures the co-occurrence significance between words with different frequencies, and normalizes it against corpus size. However, this results in assigning low frequency words with higher significance, which results in the misleading deduction that mutual dependence of less frequent words is more informative than the mutual dependence of their pairs [58]. As an alternative, Pointwise MI (PMI) is used, where MI is normalized by the plain (or log) frequency of the co-occurrence, as in

$$PMI(i, j) = freq(i, j)MI(i, j). \tag{2.21}$$

In order to limit the computed value to the positive range, e.g., for building positive weight matrices for DSMs, Positive PMI (PPMI) is typically used, essentially by assigning 0 to the case where $PMI(i, j) < 0$. In general, MI tends to rank very rare words higher in terms of significance, whereas PMI contrarily emphasizes the frequency of words.

Context metrics

The underlying hypothesis behind context-based metrics is that semantically close words tend to share the same linguistic context, i.e., similarity of context implies similarity of meaning [3]. Context-based metrics utilize a contextual window of variable size, which is centered on a word of interest i. For every instance of i, the H words left and right of i are considered, i.e., the window has a size of $2H + 1$. Thus, a feature vector is composed to represent i, that has as many dimensions as the vocabulary entries (i.e., context words). Then, a binary scheme is used in order to assign values to the vector, based on the occurrence of a context word within the contextual window, considering all occurrences of i

in the corpus. This binary scheme is typically re-weighted, and the dimensionality of the matrix that the set of vectors comprise is reduced. Then, context-based similarity metrics can be used to compute similarity between words, by utilizing their respective vectorial representations. Various context-based similarity measures have been proposed in the literature that utilize VSM modeling for similarity estimation. VSMs adhere to the geometric metaphor of meaning, i.e., words can be mapped to a semantic space with geometrical properties. Then, distance between two vectors can be estimated using metrics based on info-theoretic grounding or inspired from geometrical properties.

Cosine. Cosine similarity is the most widely-used metric, within the framework of DSMs. It estimates similarity between words i and j as the cosine of their feature vectors $\vec{i}$ and $\vec{j}$, respectively, i.e.,

$$S_C(i,j) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \|\vec{j}\|} = \frac{\sum_{k=1}^{n} i_k j_k}{\sqrt{\sum_{k=1}^{n} i_k^2} \sqrt{\sum_{k=1}^{n} j_k^2}}. \tag{2.22}$$

To make up for words that do not share a common context, similarity between them is assigned 0.

Manhattan-norm. The Manhattan-norm metric (MN) measures semantic distance as the divergence of the bigram distributions at the left and right context of each word. To this end, it utilizes the Manhattan metric (also referred to as *City-block*, or *L1 metric*) [59] to measure semantic distance as the absolute value of those distributions difference. The metric is bounded and symmetrical, since the Manhattan metric is bounded itself and considers both normal and inverse word order distributions, i.e., [60]

$$D_{MN}^{LR}(i,j) = D_{MN}^{L}(i,j) + D_{MN}^{R}(i,j). \tag{2.23}$$

Euclidean. Another metric used for measuring semantic distance is the Euclidean distance metric (also referred to as *L2*), defined as

$$dist_{L2} = \sqrt{\sum_{i=0}^{n} (a_i - b_i)^2} \tag{2.24}$$

Its main difference with the Cosine is that it considers and is influenced by vector length, i.e., it measures positions in vector space, instead of vector directions.

Kullback-Leibler divergence. Kullback-Leibler (KL) is a relative entropy measure that considers the task of semantic similarity between two words as the comparison of their conditional distributions. Using the divergence as a distance measure, however, reveals a few complications. Although its asymmetry can be addressed by taking the sum (or mean) of both divergences, or by using reversed order context distributions [61], the fact that the distance measure is unbounded (since denominators may approach zero in many cases) is not so trivial to address, as the measure can be dominated by few terms.

Jensen-Shannon. A solution to address the unbounded problem of KL is by using the total divergence to the mean or with other words, as proposed by [62, 63]. The Jensen-Shannon (JS) divergence compares both distributions to the mean of the two distributions, therefore addressing both asymmetry and the unboundedness of KL.

Information-Radius. Another metric used as a similarity measure is the Information-radius (IR) distance, which is similar to JS in the sense that it also addresses unboundedness, since it considers the average of the two probabilities as its logarithmic ration denominator. IR can be made symmetric similarly to Eq. 2.23, where, for instance, $D^L_{IR}(i,j)$ is defined as [60]

$$D^L_{IR}(i,j) = \sum_{v \in V} log \left[\frac{p^L_i(v|i)}{\frac{1}{2}\left(p^L_i(v|i) + p^L_j(v|j)\right)} \right], \tag{2.25}$$

where p^L_i is the left-context conditional probability of word i (similarly for p^L_j), and V defines the context vocabulary.

Similarity measures that are defined on distributions are closely interlinked to the approach of vector computation, and the overall performance of each measure is closely related to both intrinsic and extrinsic parameters, such as corpora selection, evaluation tasks, formulation of vectors, etc [64]. The selection of the similarity measure is, thus, based on the task at hand. For the task of estimating semantic similarity between words, the cosine similarity has been reported to perform well and is the most widely-used. Cosine is typically considered as the standard choice in DSM modeling and has been adopted by most evaluation studies in the field [12, 59, 65].

2.2.6 Composition

When considering complex structures, such as phrases or sentences, one way to estimate the semantics of said structures is to apply the same models used for the estimation of semantics in words. This is referred to in [66] as the holistic, or non-compositional, ap-

proach, as intrinsic information about the composition of the phrases is entirely ignored and semantics are computed completely based on context. Yet, however efficient as DSMs may have proved to be at capturing word semantics, their limitations are revealed when considered for semantic estimation of such structures. Treating multi-word constructions as a unit and deriving semantics the same way as for semantics for unigrams, i.e., using the distributional hypothesis of meaning via the extraction of context-based features, does not scale. The number of combinations for phrases increases exponentially with vocabulary size and structure length, resulting in a respective demand for context information. This can scale out of control, especially when considering large structures. This approach is further not in line with the cognitive comprehension of phrases; we are able to infer meanings for complex structures, even though we may have not detected them within context. This intuitively leads to the hypothesis that meaning for a complex structure is inferred from knowledge that is already present within the structure's components. However, it should be noted, that there are cases when taking the holistic approach for phrases is a sensible method, such as for estimating semantics for phrases that make use of figurative speech, or for high frequency n-grams. It is also useful to extract context-based (i.e., observed) information to compare with semantics derived by other approaches, or for learning functions that can approximate such semantics.

Compositionality allows for the construction of complex meanings through combining semantics of simpler elements. Linguistic structures adhere to the notion of "compositionality", i.e., complex structures are formed by the combination of simple linguistic entities that represent concepts and operations. Phrases are composed by words, sentences are formed from combinations of phrases, and so on. The intuition for *compositional approaches* is, subsequently, that meaning also adheres to the same principles: the meaning of a complex linguistic structure can be derived via a function that combines the meanings of its constituents. Regardless of the best approach in estimating semantics, modeling the meanings of complex phrases involves also modeling the way in which those meanings are combined. Syntactic relations also encode salient information regarding the underlying function between the constituents. What is more, a composed meaning may also account to something larger than the combined meaning of its parts. Frege's principle of compositionality suggests that the meaning of a sentence should be explained as the meaning of its words in isolation, along with the meaning that derives when combining them into sentence parts [67]. Lakoff [68] further supported this idea, proposing that the meaning of the whole can be greater than the meaning of the parts. The above suggest that the meaning of a

complex structure could encompass semantics that may not be directly estimated or accessed through its constituents, when considered in isolation. This kind of knowledge does not solely derive from word lexicalizations and the syntactic relations between them; it also incorporates our subjective sensory interpretations of the world, modified by our previous experiences, such as memories, thoughts and deductions, i.e., the compositional process involves the way that novel interpretations integrate with existing knowledge. These observations where modulated in [11] under a function that acts on the meanings of two constituents, i_1 and i_2, to derive the meaning of a complex phrase, $i = (i_1\ i_2)$, such as that

$$i = f(i_1, i_2, R, K), \tag{2.26}$$

where R is an argument representing the syntactic relation between i_1 and i_2, and K is an argument representing all knowledge that is activated during the compositional process. Another issue that becomes apparent when estimating compositional semantics is that compositionality is also a matter of degree. Some complex linguistic structures appear to derive their meaning by directly combining the meaning of their parts. Consider the bigram "red triangle": the composed meaning in this case is just the intersection of "everything that is red" and "everything that is triangle", i.e., "red triangle" acts as a fully compositional structure. On the other hand, some linguistic structures, such as idioms, i.e., phrases that make figurative use of language, or some multi-word expressions, such as, e.g., "piece of cake", "put to the sword", etc., behave as non-compositional, since their meaning is more accurately approximated by holistic approaches, i.e., information about their context contributes more than the meaning of their parts. In [69] it is proposed that the meaning of a word should not be considered in isolation, but based on the linguistic environment of the statement, i.e., the composed meaning is estimated from the constituents, and the meaning of said constituents is derived from the composition. There is no criterion to categorize linguistic structures into fully compositional or fully non-compositional; some structures appear to be just partially compositional, as are, e.g., some syntactically fixed structures (e.g., "take advantage"), in which case each of the constituents has a unique effect on the structure's meaning [70]. Compositionality appears to be a matter of degree rather than a binary concept. VSMs are typically used to represent words in isolation, and meanings are encoded into vectors spaces. Various approaches in the literature regarding composition propose different approaches to derive compositional semantics by combining the constituent vectors through a specified function. In the rest of the section we describe some of the most widely accepted approaches.

Vector Composition. The most straightforward method for deriving a compositional representation for a complex structure is to merely combine the constituent vectors. In [10, 11], various methods have been proposed that focus on such compositions. One of the simplest set of methods regarding vector composition are those decribed by the **additive model**, i.e., regarding the composed vector of a complex structure as a linear function of the cartesian product of its constituent vectors [46]. To this end, a bigram $i = (i_1\ i_2)$ can be represented by a vector $\vec{i}$ that is computed as

$$\vec{i} = A\vec{i_1} + B\vec{i_2}, \tag{2.27}$$

where $\vec{i_1}$ and $\vec{i_2}$ are the vectorial representations of i_1 and i_2, respectively, while A and B define matrices which determine the contributions made by $\vec{i_1}$ and $\vec{i_2}$ to $\vec{i}$. In its simplest form, $A = B = I$, which results in $\vec{i}$ being computed by a simple point-wise addition of $\vec{i_1}$ and $\vec{i_2}$'s values, i.e., $\vec{i} = \vec{i_1} + \vec{i_2}$. This method blends together the content of the constituents, but, although effective at some tasks, the scheme's lack of considering syntactic structure computes identical representations for structures sharing the same vocabulary. For example, the bigrams "business world" and "world business" have different meanings, i.e., the first describes a world devoted to business while the second describes business around the world. However, the sum of their vectors will result in an identical representation for both cases. To this end, the simple additive model proved ineffective, at least for the task of semantic similarity where word order plays a vital role. This is further enhanced if we consider longer and more complex structures, such as sentences, where the subject might become the object and, respectively, a causer might become the receiver of an action. Although this simple point-wise addition is word-order insensitive, using A and B can be used to weight each constituent's impact in the composition. In [71], the additive model is extended to incorporate semantic neighbors as a means of representing background knowledge in

$$i = i_1 + i_2 + \sum_{i}^{k} n_i, \tag{2.28}$$

where $n \in N$ defines a member of a set of neighbors N of k size. The author composes N by selecting the m most similar neighbors to the first constituent, refining them by the k most similar to the second, i.e., N consists of a subspace of the predicate i_1 neighborhood that is closer to argument i_2. The aforementioned types of composition do not consider the impact that a constituents' contribution has in the other. A proposed method of combining

vectors in order to give more emphasis to the components that are more relevant between the two constituents is the **multiplicative model**. The model computes the composed vector through the tensor product of its constituents and is defined as

$$\vec{i} = C\vec{i_1}\vec{i_2}, \tag{2.29}$$

C being a 3-rank tensor, which projects the tensor product of $\vec{i_1}$ and $\vec{i_2}$ onto the space of $\vec{i}$. In its simplest form, $C = I$ and Eq. 2.29 is reduced to being a simple vector product $\vec{i} = \vec{i_1} \odot \vec{i_2}$, i.e., a simple point-wise multiplication of $\vec{i_1}$ and $\vec{i_2}$'s values. Variations include taking all pair-wise products of $\vec{i_1}$ and $\vec{i_2}$, i.e., $\vec{i} = \vec{i_1} \otimes \vec{i_2}$, or taking their circural convolution, i.e., compressing the tensor product by summing along its transdiagonal values.

The Dual-space model. Turney [66] argued that compositional semantics are characterized by four linguistic phenomena that need to be handled successfully by semantic models, specifically

- Linguistic creativity: the ability of language to derive a phrase consisting of an infinite number of words and combinations.
- Order sensitivity: the consideration of word order between words, which has a great impact on the derived meaning of the composition.
- Adaptive capacity: the ability of language components to form arbitrary types of syntactic relations among them, which may be ignored by the followed approach to derive semantics.
- Information scalability: the prerequisite that an increase in the number of component words in a phrase should not be associated with exponential growth in information, neither should any loss of information result as a sacrifice to cover this condition.

Turney proposed a **dual-space model** that combines relational and compositional methods for representing compositional semantics, based on the intuition that the domain and functionality of a word are characterized by the nouns and the verbs, respectively, that appear within its context. He argued that nouns are conceptually related to the target word, while verbs associate with its syntactic, functional environment. To this end, his approach utilized two complementary models, specifically, a *domain space* and a *function space*, motivated by an attempt to address the aforementioned series of phenomena. Both models are constructed using the same approach: a matrix is built by collecting context

term frequency, followed by PPMI reweighing and, subsequently, reducing the matrix's dimensionality via SVD. The difference between the two spaces resides in the context vocabulary: in the domain space, nouns were used, while, for the function space, verb-based patterns were utilized. Composing semantics was thus reduced to combining the representations from the two models. The dual-space model, therefore, does not construct a general-purpose representation for complex structures; instead, the composite meaning is computed based on the task at hand.

Transformational Models. The described models have been criticized for their use mainly as bag-of-words models, i.e., they ignore word order, and the functional effect that words apply on their linguistic environment inside the structure and the way they can change its composed meaning. For example, a "nice" table is still a table, but a "fake" or "broken" table is not. To this end, the multiplication class, defined in Eq. 2.29, can be regarded as a function where the representation of the first constituent performs a transformation on the representation of the second. In particular, Eq. 2.29 can be formatted as

$$\vec{i} = U\vec{i_2}, \tag{2.30}$$

where $U = C\vec{i_1}$ is the partial product of C with $\vec{i_1}$. In Eq. 2.30, U can be seen as a matrix representation of the first constituent. To this end, U functions as an operator on $\vec{i_1}$, i.e., it operates on the representation of the second constituent into the vectorial representation of the bigram i. This idea serves as a cornerstone in logic-based semantic frameworks [67], in which word order determines the type of function for each component. The aforementioned adaptation was proposed by [72] for the estimation of semantics, when considering adjective-noun constructions. The idea is that adjectives act as linear transformations on noun vectors, and that the representation of an adjective-noun construction can be computed by a matrix-by-vector composition of their components. For example, when considering an adjective-noun bigram, such as "bad cat", the operator word ("bad") acts as a modifier to the head word ("cat"), i.e., it modifies the latter's meaning in order to derive the composition. Baroni and Zamparelli [73, 74] built on this idea that utilizes syntax and word order in order to estimate compositional semantics. To this end, they argued that adjectives can be regarded as modifiers on the composed meaning of the structure, and proposed that adjectives act as functions, operating on the meaning of the noun, and that the representation of the composed meaning can be derived by a transformational composition. The application of these functions is realized via matrix-by-vector multipli-

cation, through the use of a transformational model, referred to as **lexical function**, and defined as [74]:

$$f(\alpha) =_{def} F \times a = b, \tag{2.31}$$

where F is the matrix-encoded function, a is the vectorial representation of the argument α, and b is the compositional vector output. The weights of F are learnt according to selected, corpus-observed, examples of input and output distributional representations. The input is the representation of the second constituent of the phrase (head word), and the output is the observed representation of the phrase, i.e., a holistic-based estimation of its vector weights. In order to learn F, the authors employed regression techniques to estimate the set of weights in the matrix so that, when multiplied with the input vector, the result will best approximate the (observed) vector output. For example, the function for the modifier *"bad"*, in the example above, can be learnt by regressing over observed vectorial representations of nouns and <*bad **> phrases, i.e., phrases that contain the modifier as the first constituent and the respective noun as the second, such as <*pet, bad pet*>, <*dog, bad dog*>, <*bird, bad bird*>, etc. In [73, 74], the authors used two regression techniques in order to train F, specifically LSR and RR. Then, the learnt F matrix weights can be applied to any noun, in order to compute the respective representation of the compositional structure. For example, using the trained set of weights for the modifier "bad", and the vectorial representation of a head noun, e.g., "cat", the composite representation for the phrase "bad cat" can be induced. Coecke [8] generalized this idea to cover all functional types, instead of just adjectives, by utilizing tensors of higher order, for example, 3-order tensors for the case of transitive verbs. The idea of a functional space also served as the fundamental basis in [75]. Here, the matrix-vector model was extended and adapted it to the computation of semantics for multi-word sequences of any syntactic type. In particular, the authors considered a linguistic structure of arbitrary length, such as a sentence, into a dependency tree, and argued that each tree node can be regarded as *both* a parameter matrix and a continuous vector; the vector being utilized to represent the inherit meaning of the constituent, while the matrix served as the function type that is activated for the constituent, when changing the meaning of adjacent lexical units. Then, the representation of sentential semantics is constructed via a bottom-up procedure, i.e., by recursively multiplying the vector of the first component with the matrix of the second, and vice versa, until the semantics of the whole structure are computed. Their proposed **MV-RNN model** utilizes a recursive neural network (RNN) that is applied in order

to learn the compositional function for computing vector and matrix representations for sequences of arbitrary length and syntactic type. Words that lack operator semantics are assigned the identity matrix, while, words that act mainly as modifiers are assigned vectors that are close to zero. Their model has combined different theoretical properties with good performance on several empirical tasks, while also generalizing several models in the literature [72–74, 76]. In [50], the authors also proposed a generalization of the learning method introduced by Baroni and Zamparelli for computing higher rank tensors, allowing for the induction of semantic representations for larger linguistic structures.

Chapter 3

Composition in NDSMs

Network Distributional Semantic Models (NDSMs) are based on the notion of semantic priming [18]. Semantic priming refers to the cognitive concept of improving performance, in terms of speed or accuracy, of responding to a stimulus, such as word or picture. Priming is cognitively motivated, as it is triggered by associating meanings with sensory interpretations and related concepts. By considering a semantic network composed from a lexicon and the associations among its entries, activational priming can be regarded as the equivalent action of a stimulus *activating* a specific sub-area of the network, whose members share similar semantic and associative properties. Based on this idea, Iosif and Potamianos [16, 77] proposed an unstructured approach for the construction of DSMs which integrates cognitive thinking with semantic theories. In their work, it is argued that the traditional approach for computing VSMs can be extended into a two-tier system, specifically by

1. encoding corpus statistics parsimoniously in a semantic network, and
2. shift similarity computation, from corpus-based techniques to functions over network subspaces (semantic neighborhoods).

Semantic neighborhoods comprise of lexical units that are semantically associated with a target. The notion of "lexical unit" can refer to any conceptually coherent lexical structure, spanning from words (unigrams) up to word sequences (n-grams). Semantic properties of said units are, therefore, encoded into the neighborhood, which is realised as a sub-graph of the complete network. The semantic neighborhood is an adaptation of the theoretical model of priming, in the case of a semantic network; the neighborhood acts as the activation area that is triggered upon the activation of the stimuli, which, in this case is its associated lexical unit. Estimating similarities between the members of the network is thus reduced to comparing their neighborhood properties. This cognitively motivated approach constituted a new paradigm for implementing DSMs, allowing for the consideration of different aspects regarding semantic similarity. Encoding activational priming into the form of undirected network sub-graphs enables the utilization of methods that derive from theoretical backgrounds of cognitive research, graph theory or algebra for comparing the encoded semantic

properties of words and, thus, modeling similarity. Computing neighborhoods for complex structures, however, leads to the same natural language phenomena that are described in Section 2.2.6. Activation areas for complex phrases can, thus, be computed as a function of activations of phrase constituents, in adherence with the principle of compositionality.

In the rest of the chapter, we describe the work in NDSMs. First, approaches from the literature are presented for estimating compositional semantics and similarity between complex linguistic structures via the use of a two-layer system, specifically an activation and a similarity layer. We extend this model by proposing schemes and metrics that address various compositional and natural language phenomena. Next, a novel method is proposed for estimating the transformational degree of linguistic structures by leveraging on the training error produced through a transformational model from the literature. Finally, a model is proposed that computes semantic similarity of phrases by utilizing this method to determine the contribution of different compositional models under a late fusion scheme.

3.1 Semantic Networks in DSMs

A semantic network can be defined as a graph $Q = (V, E)$ whose set of vertices V includes the lexical units under investigation and whose set of edges E contains links between the vertices. The links between the units in the network are weighted according to their pairwise semantic similarity, which can be computed using some defined similarity metric [1]. The network acts as a parsimonious representation of corpus statistics that pertains to the estimation of semantic similarities between unit pairs, as they are represented by the vertices. Given a selected lexical unit, an area is modeled within the semantic network that clusters a set of semantically related lexical units. This area is referred to as *semantic neighborhood*, and its members as *semantic neighbors*. The semantic neighborhood of a lexical unit, ξ, can therefore be regarded as a sub-graph of Q, Q_ξ, also referred to as the activation area, or activation, of ξ. The vertices of Q_ξ, i.e., the neighbors of ξ, are determined according to their semantic similarity with ξ. Given a set of lexical units, the most semantically related to ξ comprise its semantic neighbors. The neighborhood is placed within the semantic concept defined by the selected lexical unit. This is especially valid for the case of highly coherent lexical units, such as unigrams and bigrams. The notion of semantic neighborhoods adheres to psycholinguistic analysis regarding the lexical interpretation of concepts, i.e., the idea that each lexical unit triggers, or activates, re-

[1] the most salient ones are described at Section 2.2.5

lated concepts or meanings that are shared among a subset of lexical units in the network. Semantic neighborhoods also play an important role in discovering relations that are indiscernible in raw data; diverse syntactic, semantic and pragmatic information is expected to be encoded as features of semantic neighbors. Such relations surface via the systematic covariation of compositional schemes and similarity metrics.

3.2 Compositional NDSMs

NDSMs are computed over semantic networks, in order to extract indirect relations between network edges. They constitute of two separate sequential layers, each addressing different semantic tasks via the utilization of semantic networks, in particular

1. an *activation layer*, responsible for computing semantic neighborhoods, and
2. a *similarity layer*, which operates on semantic neighborhoods to estimate similarities between their respective lexical units.

The activation layer pertains to computing an appropriate activation that can more accurately represent a target lexical unit. In the case of selecting neighbors for a simple unigram, this layer merely consists of selecting the n most similar to the latter. For the case of more complex structures, however, such as bigrams, computing activations in not as trivial. Properties of natural language, as well as compositional phenomena, can affect the computation of a representative activation area, as the relations between the constituents, as well as the syntactic properties of each word, may seriously impact the quality of the derived neighborhood. This is also applicable in the similarity layer, where some similarity metrics also consider these constituents for their operations. Thus, inherent properties of natural language, such as word order, appear to affect the computations in both the activation and the similarity stage, and need to be addressed appropriately for each case.

In order to describe the details in the activation and similarity layer of NDSMs, as well as to highlight and address the associated problems, the bigram lexical unit will be used as a use case. To this end, we define two bigrams, e.g. $i = (i_1\ i_2)$ and $j = (j_1\ j_2)$, where i_1 and i_2 denote the first and second constituent of bigram i (similarly for j). Also, let N_{i_1}, N_{i_2}, N_{j_1} and N_{j_2} define the activations of i_1, i_2, j_1 and j_2, and let N_i and N_j define the composed activations of i and j, respectively. Since the constituents of i and j are unigrams, their neighborhoods are easily computed as described in Section 3.1, i.e., by selecting the n most similar words to the unigram. N_i and N_j, however, are computed as

a function of their respective neighborhoods. Following this paradigm, we adapt the ideas regarding NDSMs, introduced in [16, 77], to the case of more complex structures [17].

3.2.1 The Activation Layer

In the framework of DSMs, neighborhoods were computed in the activation layer for the case of unigrams in [16], and were extended to short phrases (bigrams) in [77]. In both cases, unigrams are selected as the members (neighbors) of the activation areas (neighborhoods). NDSMs typically form unigram activations by selecting the semantically closest words to comprise their semantic neighbors. For the case of unigrams being the targeted lexical units, the activation layer merely comprises of the semantic neighborhood of the unigram. In [77], it was proposed that composed semantic neighborhoods can be computed for a complex structure, specifically for bigram, noun-noun structures. The approach was based on the intersection of senses; given a bigram $i = (i_1\ i_2)$, the semantic neighborhood of i is computed by taking the neighborhood overlap of i_1 and i_2, i.e., N_i is composed by selecting the neighbors that are shared between the activations, as

$$N_i = N_{i_1} \cap N_{i_2}. \tag{3.1}$$

This is illustrated with an example in Fig. 3.1. In this example, N_{i_1} ={"circumflex", "city", "market", "province", "investment", "region", "partners", "fund", "finance", "punishment"}, while i_1 is the word "capital", and N_{i_2} ={"stock", "investment", "global", "estate", "fund", "finance", "industry", "analysis", "value", "trading"}, while i_2 is the word "market". The composed neighborhood for the bigram "capital market" (i) is computed as N_i ={"investment", "fund", "finance"}. This approach pertains to the idea that the meaning of a complex structure should be more specific than the meaning of its parts [78]. However, it does not consider specific phenomena which prevent scaling the model to more complex structures, while it ignores important properties of compositionality. Applying the described model to the case of longer structures makes a certain limitation apparent. The fact that the resulting neighborhoods are composed using a fixed size for the neighborhoods of its constituents (e.g., N_{i_1} and N_{i_2}) allows the computation of empty neighborhoods for a complex lexical unit (e.g., N_i), when there is no overlap between the constituents' neighborhoods. Moreover, as stated by Frege [67], the meaning of a complex structure should also be explained along with the meaning that derives from the composition, which is something that a simple intersection scheme may not detect. Last but not least, the composition

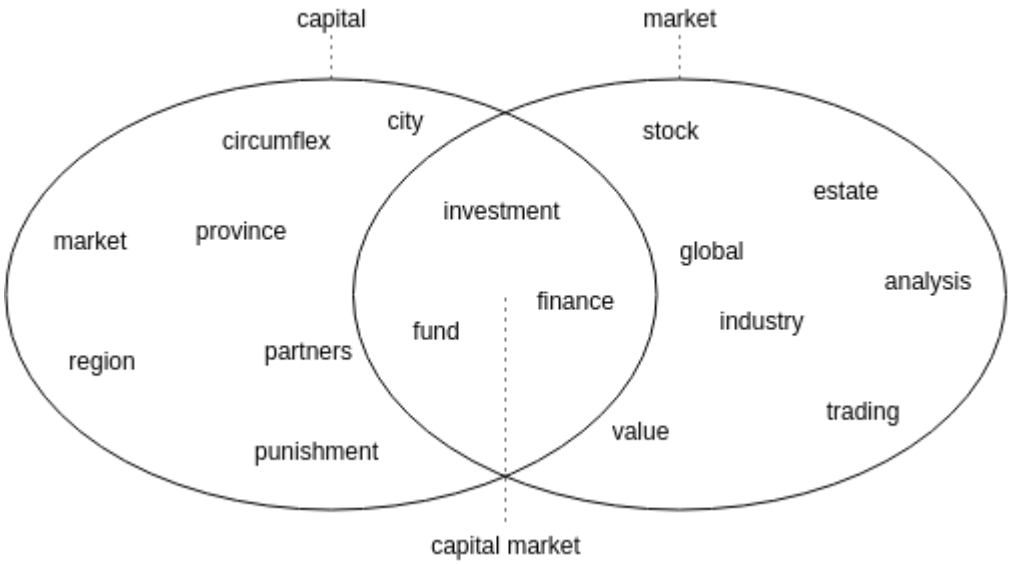

Figure 3.1: Activation composition for the bigram "capital market": The neighborhood is composed as the intersection of fixed-size constituent neighborhoods.

function should also consider the relations between the constituents, as well as their word order, when selecting neighbors for the composed activation area; syntactic infromation should affect the contribution of an activation to the composed neighborhood. In order to address this problem, the computation of composed activations is aided by weighing the contribution of the constituent neighborhoods. To this end, let n define a neighborhood member, and $S(.)$ be a similarity metric. $S(i, n)$ is then computed as (similarly for $S(j, n)$)

$$S(i, n) = w_{act} S(i_1, n) + (1 - w_{act}) S(i_2, n), \tag{3.2}$$

where w_{act} defines a weight parameter ranging between 0 and 1. By weighting the constituent contributions during the neighborhood composition, the composed neighborhood and, thus, its encoded semantics, can be regulated with respect to each phrase and based on the impact of its constituents on the composed meaning, as well as on the relation type between each other.

In order to address the rest of the described problems, five different schemes are presented for the computation of neighborhoods in the activation layer. The model proposed by [16, 77] differ from the model proposed in this work in the sense that the former requires predefining the size of the constituent neighborhoods, while the latter instead require setting the required size for the composed neighborhood. To this end, the literature models will be referred to as *fixed-size* models, while the models presented in this thesis, for

which the constituent neighborhood sizes are dynamically adapted, will be referred to as *variable-size* models [17].

Scheme 1: Intersection (*inter*). In this scheme, the neighborhood of a lexical unit is computed by taking the intersection *inter* of the constituent neighborhoods. For the case of bigram $i = (i_1\ i_2)$, this corresponds to $N_i = N_{i_1} \cap N_{i_2}$. This adheres to findings from the literature of psycholinguistics suggesting that the composed activation and, therefore, its respective meaning, should be more specific than those of its constituents [78]. To address the limitation caused by the size overlap of the constituent neighborhoods, an extension of the literature model is proposed. In particular, it is proposed that activation size is adapted by relaxing the hard constraint regarding the fixed size for the constituent neighborhoods. To this end, given a complex structure, e.g., $i = (i_1\ i_2)$, in order to compute the activation area N_i, the activation areas (i.e., sizes) of N_{i_1} and N_{i_2} are gradually extended until a minimum size, θ, is reached for N_i. The variability of constituent sizes is illustrated in Fig. 3.2, while a simpler intersection example is contained in Fig. 3.3. In order to distinguish the fixed- from the variable-size *inter* schemes, the former is annotated as $inter_{fix}$, and the latter as $inter_{var}$.

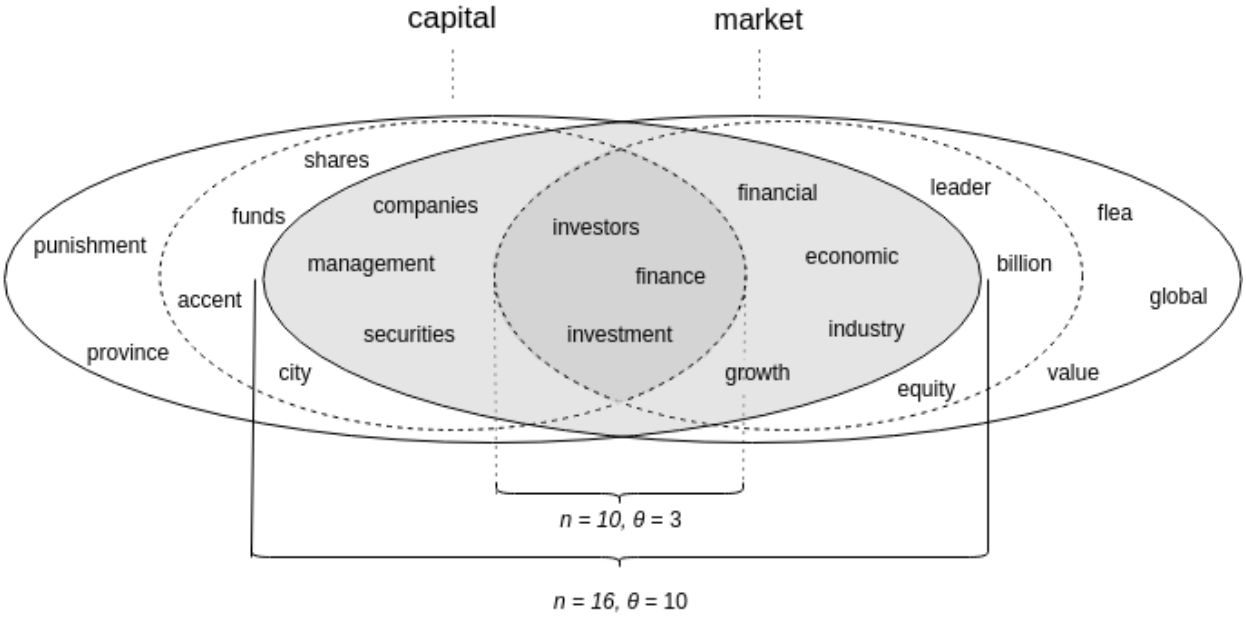

Figure 3.2: Scheme 1 (Intersection) activation composition for the bigram "capital market": Constituent neighborhood sizes (n) are incrementally increased until a minimum size ($\theta = 10$) is reached for the composed neighborhood.

Scheme 2: Union (*union*). In this scheme, the union *union* of neighborhoods is used to compute the composed neighborhood, e.g., $N_i = N_{i_1} \cup N_{i_2}$. This scheme is motivated by the idea that, in some cases, a lexical unit should associate with a larger activation area, compared to those of its constituents. This is in line with Frege's principle of compositionality [69], which indicates that the meaning of the whole encodes information that is not present in its parts, when considered in isolation; the composition may induce semantics that are only triggered via the combination of the separate word components. An example of the scheme is contained in Fig. 3.3.

Scheme 3: Most similar (*mostsimilar*). In this activation scheme, *mostsimilar*, the members for the structure's activation are selected based on their computed semantic similarity, with respect to its constituents. Specifically, let N_i be $\{n_1, ..., n_m, ..., n_\theta\}$, where $n_m \in \{N_{i_1} \cup N_{i_2}\}$. The N_i set can be regarded as a list, ranked according to $S(i, n_m)$, where $S(.)$ stands for a metric of semantic similarity. The idea behind this scheme is that different activations may be computed for N_{i_1} and N_{i_2}, given the context of i_1 and i_2, respectively, i.e., i_2 for i_1 and i_1 for i_2. This is motivated by the notion that different semantics are triggered for a constituent word based on context (where context is, from an intrinsic perspective, the other constituent(s) of its enclosing structure) and the relations between them [69]. The scheme also addresses a scalability issue: the resulting neighborhood retains the same size as those of its constituents, enabling the recursive application of the model over longer structures. An example of the scheme is contained in Fig. 3.3.

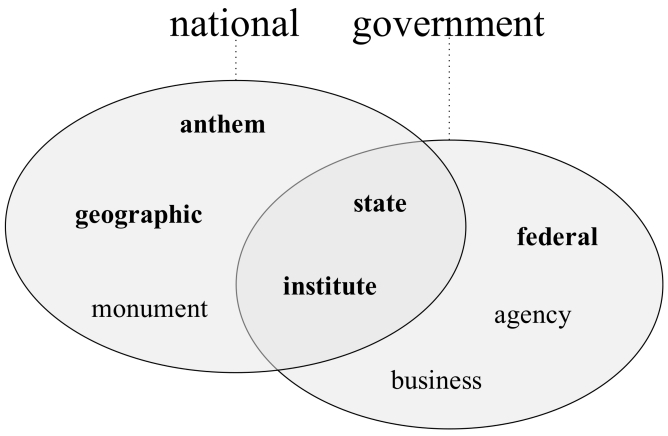

Figure 3.3: Activation schemes for the phrase "national government": intersection *inter* (overlap of neighborhoods), union *union* (members of both neighborhoods), and selection of most similar neighbors *mostsimilar* (members in bold).

Scheme 4: Head edge (*headedge*). The *headedge* scheme selects appropriate neighbors

by utilizing the network edge that connects the second (head word) with the first constituent as an inclusion radius that originates from the former. To this end, the composed neighborhood comprises of all members of N_{i_2}, and those members of N_{i_1} that are semantically closer to i_2 than i_1 is, as illustrated by Fig. 3.4. To this end, N_i is computed based

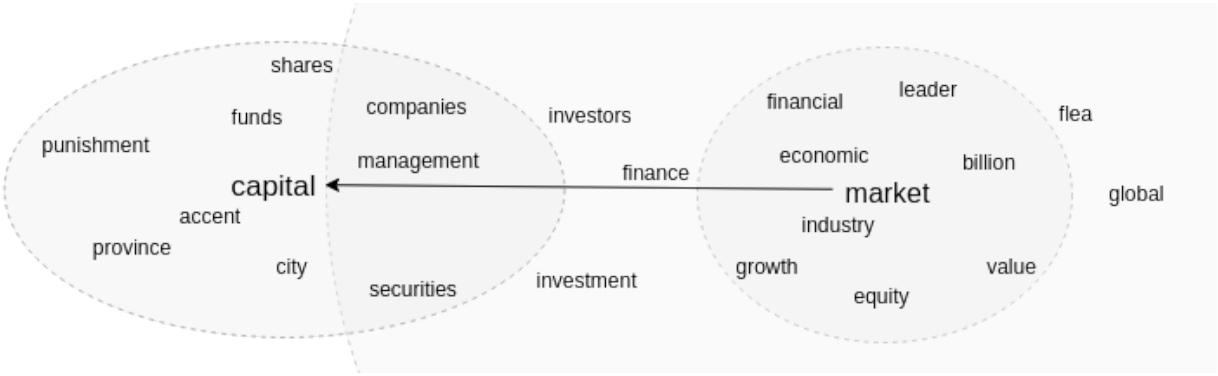

Figure 3.4: Scheme 4 (Head edge) activation composition for the bigram "capital market": Composed neighborhood is built by the activation of the head word ("market"), enhanced by the first word ("capital") neighbors that are semantically closer to the second than the first is.[3]

on the following conditions:

$$n_i \in N_i \begin{cases} \text{if } n_i \in N_{i_2}, \text{ or} \\ \text{if } n_i \in N_{i_1} \text{ and } S(n_i, i_2) \geq S(i_1, i_2). \end{cases} \tag{3.3}$$

The scheme is motivated by the idea that only specific properties of constituent neighborhoods are activated for shaping the composed meaning, based on the linguistic environment inside the enclosing phrase. Therefore, varying neighborhood properties may be activated for a lexical unit, based on the type of relation between the unit with the other constituents. This scheme composes neighborhoods based on network edge rather than activation size, i.e., the derived activation does not have a specific size, but instead uses a network edge as the criterion for member selection. The scheme also focuses on the head word activation to initialize the composed neighborhood (at least for the case of bigram structures), which is then enhanced by the inclusion of neighbors from the first word's activation. This method is in line with other work from the literature that proposes that the composed meaning of a bigram structure is estimated by an operation of the first word on the meaning of the second [72–74]. The scheme also tries to capture domain properties (encoded in the head word activation), as modified by the attributional properties (encoded in the first word

activation) that best engage it. Finally, the approach ensures that at least $|N_{i_2}|$ neighbors will compose N_i. The scheme adheres to the theory that a complex structure is not always just a sum or a direct composition of its parts, but rather a function of their relation.

Scheme 5: Projection (*project*). The scheme is based on the idea presented in [79], i.e., the hypothesis that high-dimensional spaces comprise of manifolds of very low dimensionality that are embedded in these structures. The authors base their approach on evidence from psycholinguistics analysis and cognitive science, which shows that knowledge is hierarchically clustered into conceptual manifolds comprising of closely related senses [80–82]. To this end, the authors extended this hypothesis into DSMs by constructing a hierarchical conceptual model based on the aforementioned assumptions, i.e., on the idea that a lexical semantic network can be projected into multiple low-dimensional subspaces. This scheme, *project*, utilizes this idea in order to retrieve the subspace in which the constituents of a complex structure share the maximum semantic properties (or, equivalently, the least semantic distance). In particular, this scheme projects i_1 and i_2 into the manifold of the network where $S(i_1, i_2)$ has the maximum value, S being a metric of semantic similarity. Since that subspace comprises of words that are closely related, and since the constituents share the maximum properties in that subspace, it is conclusively deducted that it encodes semantic features that are representative of both the (triggered) constituent semantics and the relational type between them. To this end, the subspace comprises the (projected) activation for the structure, i.e., N_i is composed by the members that constitute the manifold where i_1 and i_2 are most closely placed. In the case that a common subspace is not found between the two constituents, the *mostsimilar* scheme is used as a fall-back scenario. A minor difference of the *project* scheme with the work in [79] relates to the task that each work attempts to address. In [79], the model is used to measure semantic similarity between two words, while, in this scheme, the projected words are the constituents of a larger structure and the task is the estimation of the composition semantics.

3.2.2 The Similarity Layer

Activation areas, computed in the activation layer, serve as metric spaces and, since they encode semantic properties, they can be compared to estimate semantic similarity between the structures they represent. To this end, the similarity layer comprises of similarity met-

[3]The figure illustrates the case when the semantic distance between the first and the second constituent is larger than the distance of the second constituent's neighbors with it; however, all head word neighbors populate the composed neighborhood, regardless of their semantic distance from it.

rics that operate over semantic neighborhoods, i.e., it is used to model semantic similarity between two lexical units, e.g., i and j. In [16, 77], three similarity metrics were proposed for estimating similarity. These metrics were defined on top of the respective activation areas of the lexical units i and j, N_i and N_j, computed in the activation layer. The metrics adopted network-based approaches that rely on two well-founded hypotheses, namely, maximum sense [83] and attributional similarity [42]. These metrics are adopted to the case of longer structures for the bigram use case and are described in detail below.

Maximum Neighborhood Similarity (M). The key idea of this metric, M, is the computation of similarities between the constituents of lexical unit i and the members of N_j (i.e., its semantic neighbors). The same is done for the constituents of the other lexical unit, j, and the members of N_i. The similarity between i and j (e.g., between

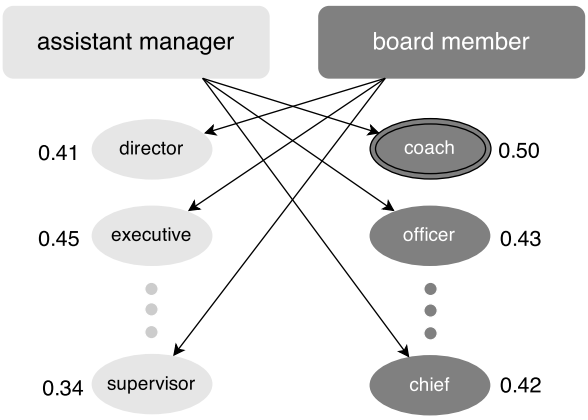

Figure 3.5: Maximum neighborhood similarity metric (M): bigram usecase.

"assistant manager" and "board member", as depicted in Fig. 3.5) is computed by taking the maximum of the aforementioned similarities (0.50 in Fig. 3.5). The underlying hypothesis is that neighborhoods encode senses that are shared between the constituents; selecting the maximum score suggests that similarity between i and j can be approximated by considering their closest senses [16].

Attributional Neighborhood Similarity (R). In this metric, R, the similarities between the constituents of i (i_1 and i_2) and the members of N_j are computed and stored into a vector. This is also done for the constituents of j (j_1 and j_2) and the members of N_j. Then, the correlation coefficient between the two vectors (e.g., the two right-most vectors in Fig. 3.6) is computed. The process is repeated, using N_i in the place of N_j, which results into another correlation coefficient (e.g., the two left-most vectors in Fig. 3.6). Similarity between i and j is then estimated as the maximum between the correlation coefficients.

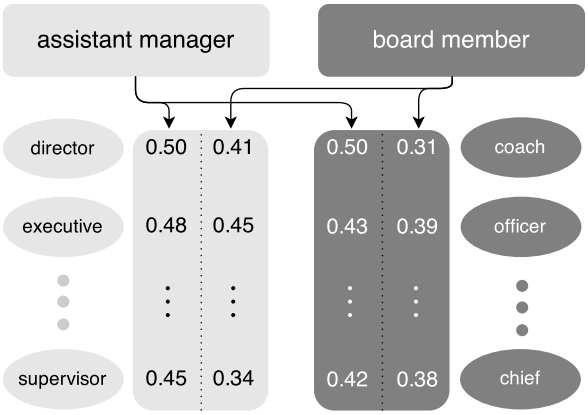

Figure 3.6: Attributional neighborhood similarity metric (R): bigram usecase.

The underlying motivation here is attributional similarity, i.e., the hypothesis that neighborhoods encode semantic or affective features; semantically similar phrases are expected to exhibit correlated similarities with respect to such features [16].

Attributional Squared Neighborhood Similarity (Q). This metric, Q, adheres to the motivation behind R, in the sense that it utilizes attributional similarity as an indicator for semantic similarity. However, the similarities among the lexical units and the neighborhood members (e.g., between i and the members of N_j) are in this case non-linearly weighted and similarity is computed as

$$Q^r(i,j) = \left(\sum_{x \in N_j} S^r(i,x) + \sum_{y \in N_i} S^r(j,y) \right)^{\frac{1}{q}}, \tag{3.4}$$

where S is a defined similarity metric. For instance, for $r = 2$, similarity between i and j is computed by summing the squares of the similarities between the members of N_i and j with those of N_j and i. This is done so that similarities contribute more to the similarity score, after weighing each member's semantic proximity with the respective lexical unit. The metric is illustrated in Fig. 3.7.

The described similarity metrics, proposed and examined in [16, 77], introduced approaches for estimating similarity via the utilization of neighborhoods. Although based on valid compositional and similarity hypotheses, these metrics were introduced for similarity estimation merely between unigrams, and were adapted to the case of simple bigram structures in order to realise a model for future reference and enhancement. Consequently, these metrics ignore some properties of language creativity, such as word order and syn-

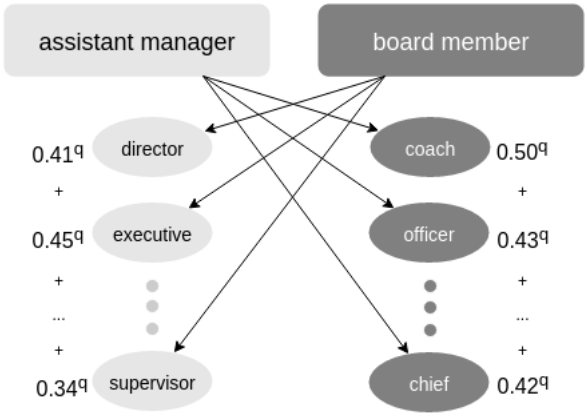

Figure 3.7: Attributional squared neighborhood similarity metric (Q): bigram usecase.

tax, rendering them ineffective when being applied to more complex and longer structures. Moreover, encoding meanings in the form of semantics placed in a metric space graph, as is the case in network-based representations, introduces properties that could be utilized by the similarity layer, e.g., in order to model similarity as the result of distance between metric spaces.

In the rest of this section, four network-based similarity metrics are presented, that attempt to address the aforementioned observations [17]. Specifically, an extension of the M metric is proposed, along with three novel similarity metrics, that compute similarity between lexical units by utilizing their activations. The metrics are defined with respect to two lexical units, i and j, which are represented by their neighborhoods, N_i and N_j, respectively, as computed in the activation layer. In order to consider the syntactic relations between the constituents, the computation of similarity is weighted in order to shift the contribution of the arguments for estimating similarity. To this end, let n be a neighborhood member, and let $S(.)$ be a defined similarity metric for estimating similarity between lexical units. Then, $S(i, n)$ is computed as

$$S(i, n) = w_{sim} S(i_1, n) + (1 - w_{sim}) S(i_2, n), \tag{3.5}$$

where w_{sim} defines a weight parameter ranging between 0 and 1. (similarly for $S(j, n)$). To this end, the described network-based similarity metrics were a special case of this generalization where $w_{sim} = 0.5$.

Average of top-k similarities (M_k). This metric, M_k, extends the M metric described above, by considering the top k similarity scores across neighborhood members and lexical units, instead of just the maximum score. Similarity between i and j is then computed

by taking the arithmetic mean of the k scores. The M metric can, thus, be defined as a special case of M_k, where $k = 1$ (M_1). The metric is proposed in order to smooth the similarity between i and j over a distribution of their closest neighbors, instead of relying on just the maximum similarity, which can produce unstable behavior.

Average of top-k pairwise similarities (P_k). This metric, P_k, estimates similarity between i and j in an indirect way, by comparing their activation areas. In particular, let C be a ranked list, including all the pairwise similarities computed between the members of N_i and N_j as

$$C = \{ \underset{\substack{x \in N_i \\ y \in N_j}}{S}(x, y) \ \}, \tag{3.6}$$

where $S(.)$ stands for a metric of semantic similarity. Similarity between i and j is then estimated as the average of the top k pairwise similarities between the neighborhood members, which is defined as

$$P_k(i, j) = \frac{1}{k} \sum_{l=1}^{k} c_l, \tag{3.7}$$

where c_l is the l-th member of C.

Hausdorff-based similarity (H). This metric, H, utilizes metric space algebra in order to compare activation areas according to their structure. The metric is motivated by the Hausdorff distance [84], defined as

$$d_H(X, Y) = \max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\}, \tag{3.8}$$

where X and Y are two non-empty subsets of a metric space (M, d), sup is their supremum and inf is their infimum. Assuming the bigrams $i = (i_1\ i_2)$ and $j = (j_1\ j_2)$, with N_i and N_j being their respective activation areas, let

$$h(N_i, N_j) = \min_{x \in N_i} \max_{y \in N_j} S(x, y), \tag{3.9}$$

where $S(.)$ is a semantic similarity metric (similarly for $h(N_j, N_i)$). Eq. 3.8 can then be adapted to compute the similarity between i and j as

$$H(i, j) = \max\{h(N_i, N_j), h(N_j, N_i)\}, \tag{3.10}$$

Euclidean-based similarity (E). This metric, E, assumes that neighbor similarities with their target lexical unit can be encoded into vectorial features. Then, semantic similarity between i and j can be estimated as the semantic divergence between two activation areas, by computing the euclidean distance of their respective vectorial representations. Let $N = (N_i \cup N_j - N_i \cap N_j)$ be a set including all unique neighbors of N_i and N_j, $n_k \in N$ being the k-th member of N, and let $S(.)$ be a metric of semantic similarity. Then, vectors for i and j of $|N|$ size are formed and populated using the similarities of i and j with the members of N, i.e., $S(n_k, i)$ and $S(n_k, j)$ similarities are encoded as vectorial features of i and j, respectively, for each $n_k \in N$. The euclidean distance between the two vectors is then defined as

$$E^{dist}(i,j) = \sqrt{\sum_{n_k \in N} (S(n_k,i) - S(n_k,j))^2}. \tag{3.11}$$

Based on 3.11, E is computed as

$$E(i,j) = \frac{1}{1 + E^{dist}(i,j)}. \tag{3.12}$$

An indirect property of this metric is that it encodes neighborhoods into vectorial representations, which provides potential use in other applications and tasks. Since VSMs constitute the most common and typically used implementation for DSMs, many well-established metrics and theories can be applied to them.

3.3 Fusion of Compositional DSMs

As described in previous parts in the current thesis, linguistic structures differ with respect to the forces that contain the composition, which derive from linguistic or syntactic properties and the relations that bind the constituents into a complex structure. Compositional models that are present in the literature and are described in Section 2.2.6 attempt to explore different aspects in compositional approaches. Although some may perform better than others in appropriate tasks, what is actually important for compositional models is to be able to capture all those different salient forces that drive the functions for each composition. The Network DSM, presented in the previous section, computes activation areas for complex phrases, such as phrases, by utilizing variable areas of their constituents' semantics under a variety of functions. The lexical function (*lexfunc*) model, defined by Eq 2.31, also described as a transformational model, utilizes the functional power of a mod-

ifier over its enclosing phrase, in order to estimate the composed meaning by operating over its semantics. Both models seem intuitively aligned with the human process of phrase comprehension, as they are based upon valid theoretical concepts of cognitive research and linguistic analysis. However, there are cases where one model fits better than the other. For example, consider two bigram phrases, "successful engineer" and "red triangle". The transformational model is expected to perform better for the first bigram, since, in this case, the composed meaning is derived from the meaning of the word "engineer", as modified by the functional influence of the word "successful". For the second phrase, however, a simple intersection of word senses, such as an NDSM using an intersection activation scheme (as defined in Section 3.2.1) or a simple additive model (as defined in Section 2.2.6) seems to be more appropriate; the composed meaning derives from the overlap of "all that is red" and "all that is triangle".

3.3.1 The Transformative Degree of Modifiers

Based on the aforementioned considerations, intuitively, complex structures can be quantified based on the relational types of their constituents, in order to derive the composed meaning. Given the syntax of the English language, it can be deducted that, at least when considering bigram structures, the first word can be considered as fitting a modifier role, while the second as being the head word of the structure, upon which the modifier operates to form the composed meaning. Such considerations have been proposed in the literature for the case of adjective-noun [73, 85], noun-noun bigrams [86], or any functional type [8], so it is safe to propose that any bigram can be defined as a modifier-head composition, while the transformational properties of the word that resides in the modifier slot of the structure can be quantified by a defined process. Measuring the transformational power of the modifier on a given head word would provide a means for detecting the relational type between the constituents, at least when considering bigram structures. This would allow for greater flexibility which will eventually benefit performance. To this end, we propose that the transformative degree of the modifier can be considered as the criterion for estimating a structure's type of composition and, subsequently, quantify it between the range of fully *transformational* or strictly *compositional*.

The *lexfunc* model, described in Section 2.2.6, represents a phrase's modifier as a function (represented by a matrix), and employs regression techniques to learn the weights that, when combined with the head word vector, best approximate the holistic (observed) representation of the phrase. The performance of said regression, when training the modifiers,

can be indicative of the fit of the *lexfunc* model regarding each modifier and, subsequently, serves as the criterion for measuring its transformative degree. Regression performance can be measured using the MSE of training the modifiers. Training MSE can, thus, determine the degree of their transformational properties, on a given head word. Taking the MSE is a sensible approach, since regression tries to compute a close approximation to the observed vectorial phrase representations and head nouns by the means of transforming the head noun vectors. A high training error would indicate that the *lexfunc* model is a poor match for this modifier, i.e., that the modifier's transformative degree is too low to be salient for semantic tasks, while a low error would indicate that the modifier responds well to the regression strategy of the model and that its interaction with the head word stems from the transformation of the latter's meaning. We extracted modifiers from the [11]

Degree	**Nouns**	**Adjectives**	**Verbs**
High	railway	old	encourage
	labour	rural	attend
	defence	elderly	remember
	personnel	efficient	satisfy
	committee	practical	suffer
Neutral	company	various	face
	care	right	need
	community	better	cut
Low	news	new	like
	service	great	buy
	business	black	help
	world	general	use
	state	good	provide

Table 3.1: Modifier examples of high, neutral, and low transformative degree.

noun-noun (NN), adjective-noun (AN), and verb-object (VO) datasets, which is used in the literature for semantic similarity tasks, and contains a variety of bigram phrase pairs. We then ranked these modifiers based on their transformative degree, as estimated from their training MSE when using the *lexfunc* model. We employed Ridge Regression as the regression technique to estimate the MSE for each modifier. Examples of modifiers of high, neutral, and low transformative degree are presented in Table 3.1. It can be deducted that modifiers with high transformative degree tend to acquire a functional role, when being regarded as components of bigram structures. For example, when considering the bigrams "efficient machine" and "new machine", the modifier "efficient" has a greater effect on

the head noun "machine", rather than the modifier "new". A "new machine" retains the same properties of a generic machine, i.e., it is a combination of all that is "machine" and all that is "new", nevertheless, regarding bigram "efficient machine", the word "efficient" includes all optimization mechanisms of the "machine", e.g., regarding speed, load, cost, etc., i.e., the word "efficient" seems to have a greater effect on the properties of the head word.

3.3.2 The Fusion Model

Based on these considerations, a fusion model is presented [17] for estimating similarity, derived from the combination of the lexical function (*lexfunc*) model, as defined by Eq. 2.31, with the NDSMs that are described in Section 3.2. This model provides a novel approach of combining different means of compositional interpretations into a similarity metric. Intuitively, a fusion model is expected to be more adaptive and sensitive to word features and effects, when estimating compositional semantics. The fusion is aimed to model more accurately the semantic representations of complex structures, by combining the two models based on their fitness degree on a given phrase structure. The fitness degree of the structure is measured according to its modifier's performance, when used to train the *lexfunc* model. Specifically, the *Mean Squared Errors* (MSEs) from training modifiers with the *lexfunc* model are used in order to estimate the transformative degree between two phrases. The transformative degree is subsequently used as a criterion for deciding whether a strictly compositional or a transformational model is more appropriate for modeling their similarity. To this end, similarity is estimated by quantifying the model contributions in order to compute the final score.

Given two phrases, $i = (i_1\ i_2)$ and $j = (j_1\ j_2)$, let the transformative degree $T(i, j)$ for estimating similarity between i and j be defined as

$$T(i,j) = \frac{1}{2}(MSE(i_1) + MSE(j_1)), \tag{3.13}$$

where $MSE(i_1)$ and $MSE(j_1)$ is the MSE that corresponds to modifiers i_1 and j_1, respectively, as measured during training them in the *lexfunc* model. Since $T(i, j)$ is an unbounded measure, we apply a sigmoid function in order to smooth and normalize its values within the range of 0 and 1. This operation results in a quantification measure (λ), based on $T(i, j)$ and utilized to adjust the contribution degree for the two models as

defined by

$$\lambda(i,j) = \alpha / \left(1 + \mathrm{e}^{-T(i,j)}\right) - \beta, \tag{3.14}$$

where α and β serve as predefined weight parameters. The proposed fusion metric, Φ^{lf}_{net}, used for estimating similarity between i and j, can then be defined as

$$\Phi^{lf}_{net}(i,j) = (1 - \lambda(i,j))\ S_{lf} + \lambda(i,j)\ S_{net}, \tag{3.15}$$

where S_{net} and S_{lf} define the similarity scores computed by the compositional NDSM and transformational *lexfunc* models, respectively. We selected models from the NDSM framework in order to provide an alternative approach to the metric and boost its power to weight towards the appropriate strategy for each phrase pair. It should be noted that the first component of the equation can be replaced with any compositional model from the literature. Thus, in addition to the aforementioned fusion, Φ^{lf}_{net}, a fusion metric combining the transformational model and the widely-used additive [10, 11] model was also implemented. This fusion metric, referred to as Φ^{lf}_{add}, is defined similarly to Eq. 3.15, where, in this case, S_{net} is substituted by S_{add}, which defines the similarity score computed by the additive model.

Chapter 4

Experiments and Evaluation

In this chapter, we describe the experiments that were defined for evaluating the proposed models. First, the evaluation dataset and metric used for measuring model performances are described. A description of the approach that was decided follows, regarding the implementation, configuration and evaluation of the compositional NDSMs and fusion models, in their respective sections. An overall presentation of the results, including evaluations of all models, is presented next. We conclude by summarizing the findings at the final section of the chapter.

4.1 Evaluation Dataset and Metric

All of the models proposed in this thesis were evaluated on the widely-used Mitchell & Lapata [11] datasets. The datasets comprise of three sets of noun-noun (NN), adjective-noun (AN), and verb-object (VO) constructions, and have been selected for multiple studies in the literature regarding the estimation of bigram similarity. Each of the datasets comprises of 108 bigram pairs, and each pair is associated with multiple ratings by different participants. The participants rated the semantic similarity of each phrase pair in a 1 to 7 scale, where a score of 1 defines the phrases of the pair being "least similar", while a score of 7 defines them as "most similar". In the current work, the participants' scores were averaged, for each phrase pair, and the resulting score was set to serve as the gold standard rating for the pair.

Model performance was evaluated against the (averaged) human judgements by utilizing Spearman's rank correlation coefficient (ρ), defined as follows: let x_k and y_k be two similarity scores that define the semantic similarity of a phrase pair $p_k = \{i_k, j_k\}$, and let $\vec{x} = (x_1, ..., x_k, ..., x_n)$ and $\vec{y} = (y_1, ..., y_k, ..., y_n)$ define vectors of n size that hold the scores that were computed for respective n phrase pairs. ρ estimates the degree of correlation between the two vectors, $\vec{x}$ and $\vec{y}$, by converting their raw scores, x_k and y_k, to ranks, x_k^r

and y_k^r. ρ is then computed as

$$\rho = 1 - \frac{6 \sum_{i=0}^{n} d_i^2}{n\left(n^2 - 1\right)}, \tag{4.1}$$

where $d_i = x_i^r - y_i^r$ defines the difference between the two ranks for the i-th phrase pair. In order to evaluate the models, ρ estimates the correlation between the observed scores, i.e., the scores that were computed by a selected model, and the gold standard, i.e., the ratings provided by the averaged human judgements.

4.2 Compositional NDSMs

In this section, we describe the experimental procedure that was followed for implementing and evaluating the compositional NDSM models, proposed in Chapter 3. We present the followed approach for realising the semantic network, followed by the description of the appropriate parameters selected for the NDSM models. Then, model evaluations are presented for the defined experiments, as considered from the following perspectives:

1. **Activation layer**: we evaluate and compare the proposed activation schemes.
2. **Similarity layer**: we evaluate and compare the proposed similarity metrics.
3. **Similarity smoothing**[1]: we investigate the impact of changing the smoothing range.
4. **Asymmetry**: we evaluated asymmetry, as considered in both layers, i.e., activation and similarity asymmetry.

Model evaluations are superseded by discussion on the findings.

4.2.1 Semantic Network

In order to implement the proposed NDSMs, a semantic network is needed, and, to create the network, a vocabulary first had to be defined. To this end, we defined the vocabulary of the network by intersecting the English vocabulary of words, found in the dictionary used by the ASPELL spell checker[2], with the Wikipedia dump [3]. The result of the intersection

[1] This perspective only concerns the metrics for which the smoothing parameter applies, i.e., the value of k for the M_k and the P_k metrics.

[2] `http://www.aspell.net/`

[3] As of the 4th quarter of 2012.

was an English vocabulary, consisting of approximately 135,000 word entries. The vocabulary entries (in this case, unigrams) were used to represent the network nodes. In order to compute network edges (vertices), the pairwise similarities among the vocabulary's entries were used. To estimate said similarities, a corpus-based approach was followed in order to gather contextual information about the vocabulary: First, for each vocabulary entry, approximately 1,000 document snippets were downloaded from the web and merged into a corpus. Then, word similarities were computed among all the vocabulary entries. This was realized by utilizing corpus co-occurrence statistics and, specifically, the Google Semantic Relatedness (S_G) metric, described in Eq. 2.19. For computing D_G (described in Eq. 2.18), which serves as a component of estimating S_G, we defined the corpus unit D at the sentence level. This convention was in line with the definition for the co-occurrence of words, which was also realised at sentence level. The S_G metric was selected after considering its good performance in word-level semantic similarity tasks [16]. As a final step for comprising the semantic network, only the top 20,000 frequent entries were considered, as a means of keeping the words for which there is more coherent information and for refining the DSM.

4.2.2 Model Configurations

NDSMs require setting a variety of parameters that relate to both the activation and the similarity layer. Configurations that were used for the experimental setup of NDSMs are presented below.

Activation Layer. Regarding the activation layer, an important parameter concerns setting the neighborhood size. Due to its use as a minimum threshold parameter in variable-size models, as opposed to its use in fixed-size models as the determiner of the constituents' activation size, the parameter has been set different values for the two variations. Specifically, let θ_{fix} define the size for the fixed-size models, i.e., the size of the constituent neighborhoods before the creation of the composed neighborhood, while, let θ_{var} define the size for the variable-size models, i.e., the minimum required size for the composed neighborhood. To this end, we defined θ_{fix} in the range of $\{10, 25, 50, 100, 150, 200, 500\}$ and θ_{var} in the range of $\{1, 5, 10, 15, 20, 25, 30, 40\}$. An additional parameter utilized in the activation layer is the activation symmetry weight w_{act}, used by the *mostsimilar* and *project* activation schemes. We defined w_{act} in the range of $\{0.10, 0.25, 0.33, 0.50, 0.67, 0.90\}$.

Similarity Layer. In the similarity layer, M_k and P_k make use of a smoothing parameter k. We set k in the range of $\{1, ..., 5\}$. The respective similarity symmetry weight, w_{sim}, utilized by the M_k, P_k, H and E metrics, was defined similarly to w_{act}, i.e., $w_{sim} =$

$\{0.10, 0.25, 0.33, 0.50, 0.67, 0.90\}$.

Baseline. In order to model the baseline NDSM and use it for comparisons with our models, we used the $inter_{fix}$ activation scheme, for the activation layer. For the similarity layer, we used the M similarity metric, i.e., the special case of M_k where $k = 1$, and set $w_{act} = w_{sim} = 0.50$. This is one of the models that was used in [77], where the model was evaluated on noun-noun bigram compositions. This model was selected as it was the best performing model on those experiments, reaching peak performance when $\theta_{fix} = 150$ [4]. In our experiments, however, we used different values for θ_{fix} (defined above), in order to investigate model performance over a variety of activation sizes.

4.2.3 Evaluation Results

Model performances are presented in this section, on the defined evaluation datasets. Due to the large set and range of the parameters used, performances are presented from the following perspectives:

Activation Layer. The performance of each of the proposed activation schemes is displayed by composing appropriate models using the similarity metrics, as well as the value of k (for the similarity metrics it applies to), that best utilize each activation scheme. The models are defined as symmetrical, i.e., $w_{act} = w_{sim} = 0.50$, then the effect of shifting the symmetry of the activation layer (i.e., changing w_{act}) is illustrated for an appropriate activation scheme.

Similarity Layer. The performances of similarity metrics is presented via two viewpoints:

1. The performance of each similarity metric is illustrated, by composing appropriate models using the activation schemes, as well as the value of k (for the similarity metrics it applies to), that best utilize each similarity metric.

2. Specifically for the similarity metrics M_k and P_k, which utilize k to define their smoothing range, we demonstrate how changing k affects each metric's performance.

The models are defined as symmetrical, i.e., $w_{act} = w_{sim} = 0.50$, then the effect of shifting the symmetry of the similarity layer (i.e., changing w_{sim}) is illustrated for an appropriate similarity metric.

[4] It should be noted that the models presented in [77] were evaluated using a subset of the NN evaluation dataset, i.e., on 92 of the 108 phrase pairs of the NN dataset. However, even when using the full set of phrase pairs, the model still performed best when $\theta_{fix} = 150$.

Activation Layer

To measure the performance of the proposed schemes in the activation layer, we experimented by combining them with every similarity metric. The models that best fit each activation scheme were selected for illustrating their performance, in order to compare among the best utilizations of the activation schemes. Information regarding the similarity metrics that best fit each activation scheme is displayed in Table 4.1. To this end, model

Activation Scheme	**Similarity Metric**		
	NN	**AN**	**VO**
$inter_{var}$	M_5	M_1	M_1
union	M_5	P_1	E
mostsimilar	M_2	M_5	M_2
headedge	M_5	M_1	M_5
project	M_2	M_5	M_2

Table 4.1: Similarity metrics that best fit each activation scheme, regarding model performances on NN, AN, and VO phrase pairs.

performances in the activation layer is illustrated in Fig. 4.1 for the case of NN, AN, and VO phrase pairs. In this case, the models are fully symmetrical, i.e., $w_{act} = w_{sim} = 0.5$.

Activation schemes. For the case of NNs, the *mostsimilar* and $inter_{var}$ schemes seem to perform best in the full range of θ_{var}, with the *mostsimilar* scheme reaching peak performance during the lowest range of θ_{var} ($\theta_{var} = 5$). However, although the *mostsimilar* scheme appears to perform best in the case of NNs, the $inter_{var}$ scheme has the most robust performance across the evaluation sets. The *union* scheme seems to perform best in the NN case, which is consistent with the hypothesis that, in some cases, the meaning of the whole might activate a larger activation area than the simple overlap of that of its constituents. Its low performance in AN and VO constructions indicates that NN relations form conceptual domains that affect a larger activation area, i.e., the meaning of NN constructions may diverge from the meaning of their isolated constituents. The performance of the *headedge* scheme appears to correlate with neighborhood size, converging towards its peak as θ_{var} increases. This could indicate that the performance of the scheme is related to neighbor position on the cartesian space; larger activations provide more members for the composition that are placed in-between the two constituents, i.e., sharing more semantic properties with both of them.

Activation asymmetry. Performance of activation asymmetry is illustrated in Fig. 4.2,

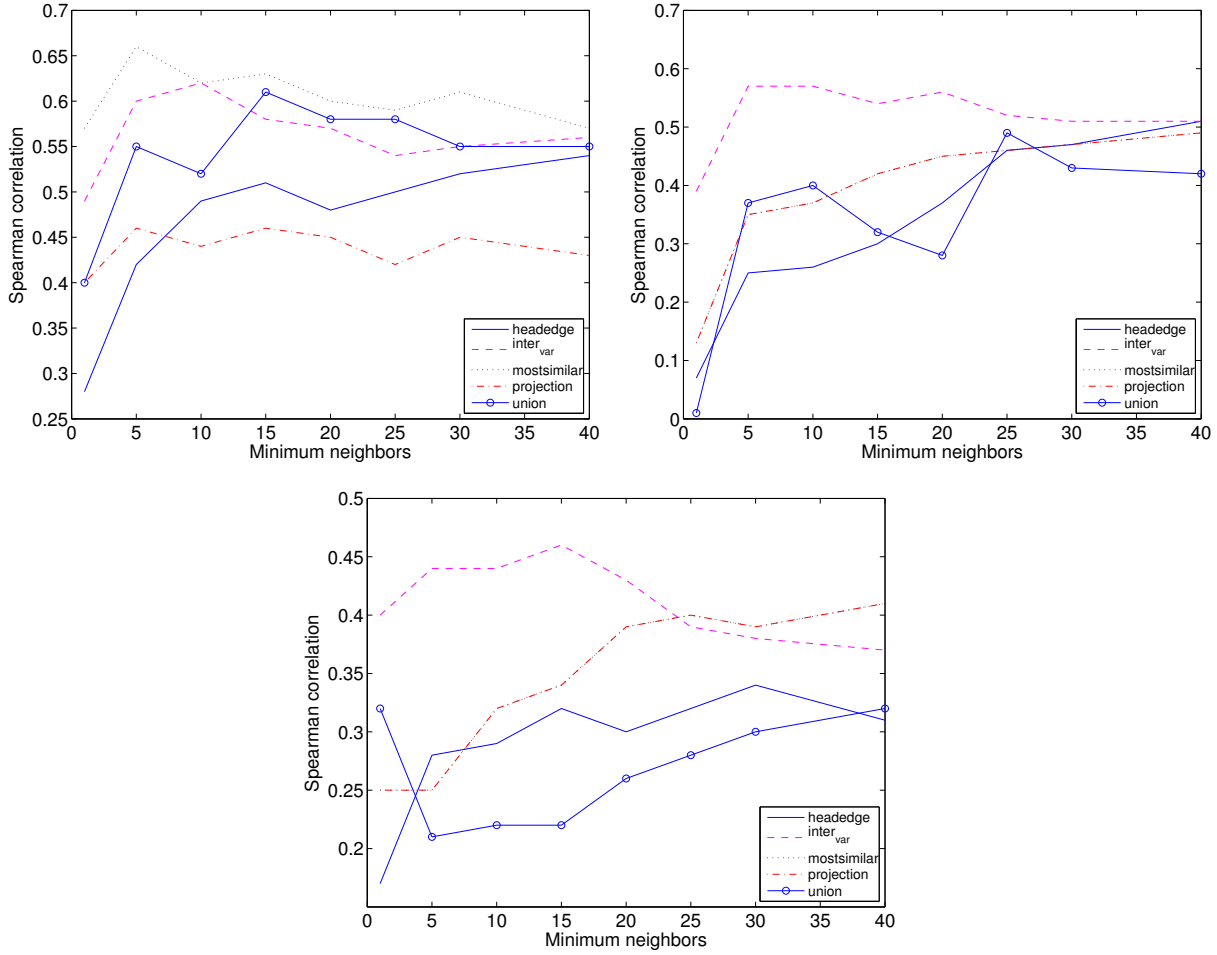

Figure 4.1: Performance of activation schemes on a) noun-noun, b) adjective-noun, and c) verb-object datasets, using symmetry models ($w_{act} = w_{sim} = 0.5$) and the similarity metrics described in Table 4.1.

for the case of the *mostsimilar* scheme, where the lines represent the values of w_{act}. In this case, models are only symmetrical in the similarity layer, i.e., $w_{sim} = 0.5$. It is confirmed that word order plays an important role for all cases, that syntactic information activates different features for each constituent and that relations between constituents adhere to their intrinsic linguistic environment. Also, in general, model performances are consistent with respect to one another as neighborhood size increases, indicating that changing the activation area does not affect the impact of word order. Finally, weighing more the contribution of the first word activation generally leads to better performance, which provides an indication of the functional influence of the first word as a modifier.

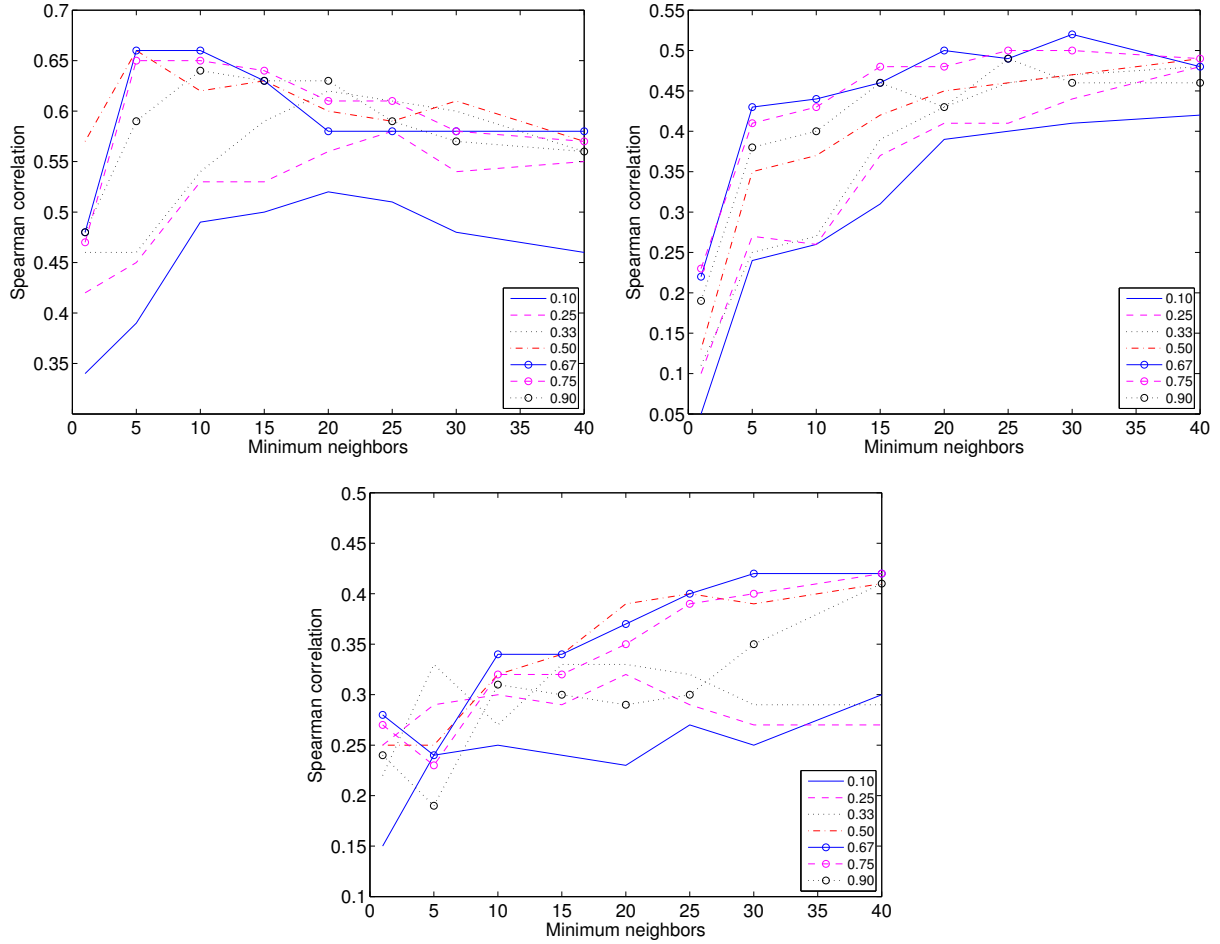

Figure 4.2: Performance of activation asymmetry (*mostsimilar* scheme) on a) noun-noun, b) adjective-noun, and c) verb-object datasets, using similarity symmetry ($w_{sim} = 0.5$) and the similarity metrics described in Table 4.1.

Similarity Layer

A similar approach was followed for the evaluation of the proposed similarity metrics. In particular, in order to measure performance, we composed models by combining each similarity metric with every activation scheme, using the whole range of k (where applicable). The models that best fit each similarity metric were selected for illustrating their performance, in order to compare similarity metrics while in their best utilization. The activation schemes and the k value (where applicable) that best fit each similarity metric, are displayed in Table 4.2. Model performances in the similarity layer are presented in Fig. 4.3, regarding each of the proposed similarity metrics, for the case of a) NN, b) AN, and c) VO phrase pairs. In this case, models are fully symmetrical, i.e., $w_{act} = w_{sim} = 0.5$.

Sim. Metric	NN		AN		VO	
	k	**Act. Scheme**	**k**	**Act. Scheme**	**k**	**Act. Scheme**
M_k	2	*mostsimilar*	1	$inter_{var}$	1	$inter_{var}$
P_k	3,4,5	*mostsimilar*	1,2,3	*union*	2,3	$inter_{var}$
Q	-	*mostsimilar*	-	$inter_{var}$	-	$inter_{var}$
H	-	*inter*	-	$inter_{var}$	-	$inter_{var}$
E	-	*mostsimilar*	-	$inter_{var}$	-	$inter_{var}$

Table 4.2: Activation schemes and the value of k that best fit each similarity metric, regarding model performances on NN, AN, and VO phrase pairs.

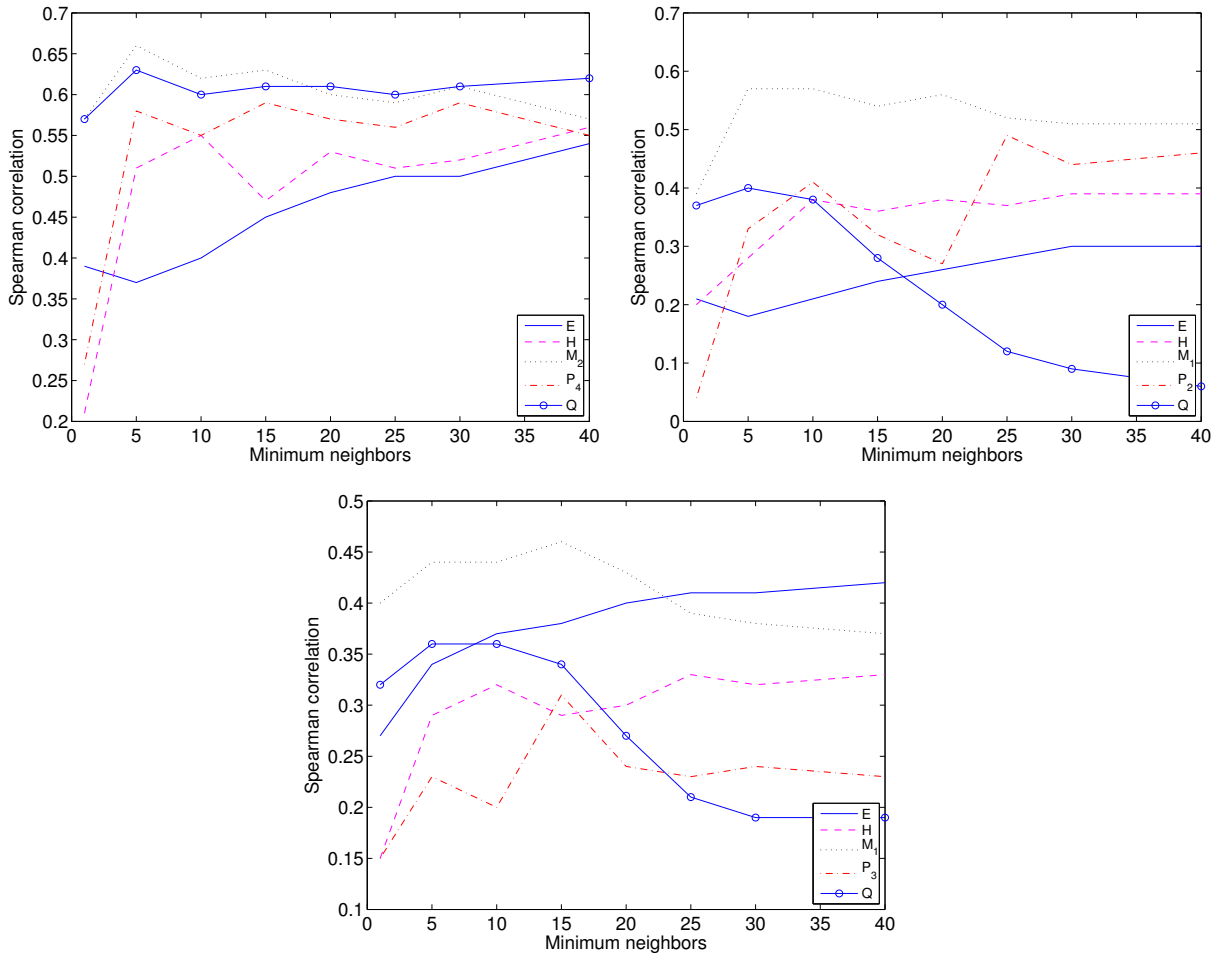

Figure 4.3: Performance of similarity metrics on a) noun-noun, b) adjective-noun, and c) verb-object datasets, using symmetry models ($w_{act} = w_{sim} = 0.5$) and the activation schemes described in Table 4.2.

Similarity metrics. Performance of the M_2 metric appears to be the most robust across the evaluation sets, while at the same time performs best than the rest of the metrics, in the majority of the experiments. For the case of NNs, both the Q and the P_4 perform almost as well as M_2. For the case of ANs, only the M_2 seems to perform well, while, for the case of VOs, the metrics adhere to the same relative performance, with the exception of the E metric that performs best for large activation sizes, even though it under-performs when applied to the NNs and ANs data sets. This suggests that vector magnitude plays a role when modeling similarity between structures that involve verbs.

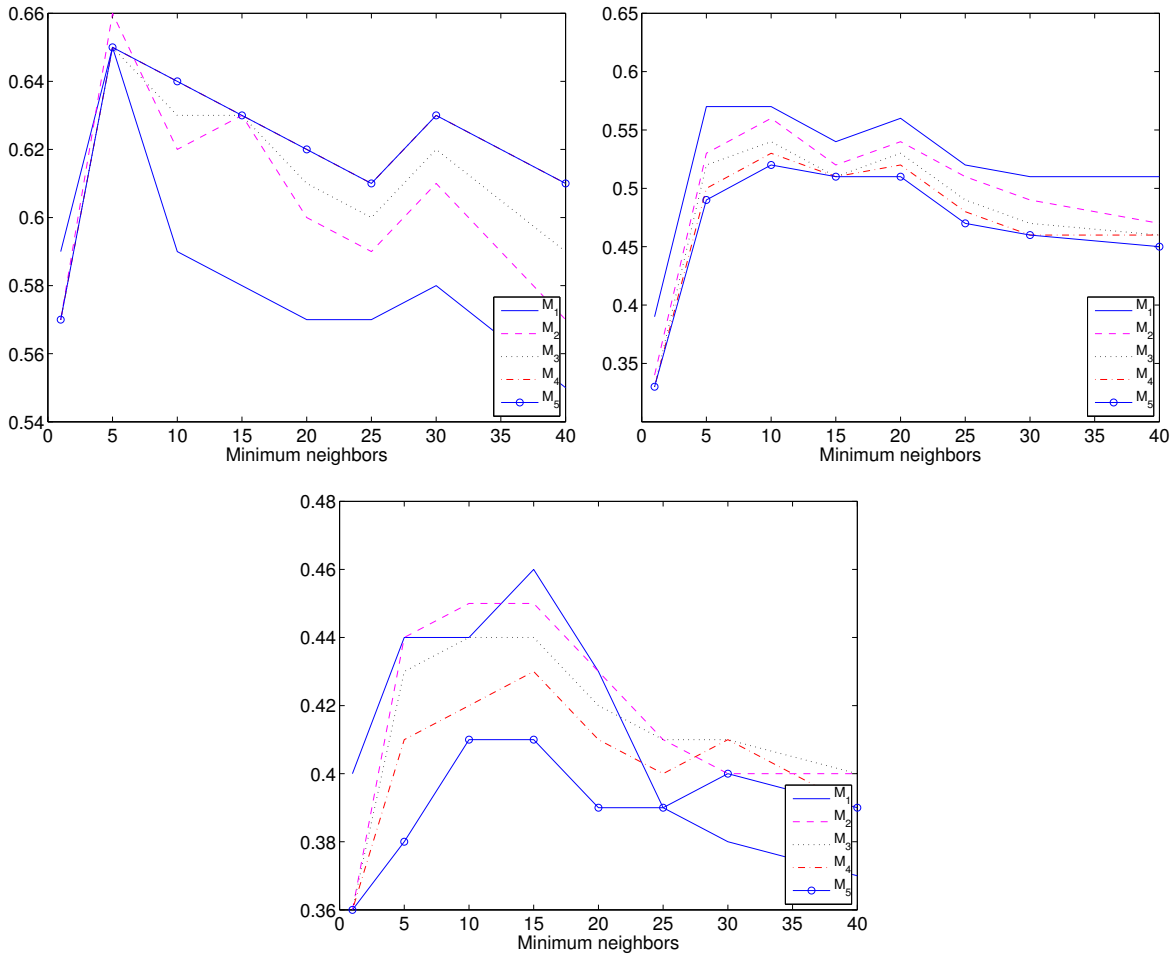

Figure 4.4: Performance of smoothing M_k with $k = \{1, 2, 3, 4, 5\}$ on a) noun-noun, b) adjective-noun, and c) verb-object datasets, using symmetry models ($w_{act} = w_{sim} = 0.5$) and the activation schemes described in Table 4.2.

Similarity smoothing. Model performances, when changing k, are presented in Fig. 4.4, for the case of NN, AN, and VO phrase pairs. Altering k in P_k has the same effect as

in M_k, so only performances for the M_k metric are presented. Once again, the presented models are fully symmetrical, i.e., $w_{act} = w_{sim} = 0.5$.

Changing the smoothing range for the metrics results in a small increase in performance for the case of NNs, confirming that activation areas encode semantic features in their range that may not be fully captured by just one specific element, and that using a wider smoothing range may infer semantic properties that are distributed in the activation that can be utilized to best approximate similarity between phrases. For the case of ANs and VOs, however, increasing the smoothing range results in an opposite behaviour, i.e., there is a respectively small deterioration in performance. Even though deterioration is statistically insignificant, this difference in behavior could be related to the fact that relations of constituents in "functional" structures, such as ANs and VOs, trigger specific semantic properties that may not be shared among all members of the composed activation, i.e., neighbors act as semantic features that are activated according to the usage of the constituent, with respect to its context.

Similarity asymmetry. Last, model performances on similarity asymmetry are illustrated in Fig. 4.5 for the case of the M_2 metric, where lines represent different values for w_{sim}. In this case, models are only symmetrical in the activation layer, i.e., $w_{act} = 0.5$. It is clear that word order has an important role, when estimating semantic similarity between phrases. For the case of NN and AN constructions, models perform best when the first word semantics contribute more to similarity estimation. For the case of VOs, however, no clear conclusions can be deducted; applying asymmetry in the similarity layer appears to have an impact on performance that, however, is not consistent in all compared phrase pairs. It seems that each phrase pair should be handled with different weights, in order to fully utilize asymmetry in the layer.

Overall

From the perspective of the activation layer, as is presented in Table 4.1, all activations schemes utilize best their functions when matched with the M_k metric, with the exception of *union* for the case of ANs and VOs (where the P_n and E metrics, respectively, seem more appropriate). This indicates that the M_k metric provides flexibility, while it is also quite robust over various activation variations and phrase types. This is confirmed in Fig. 4.3, where it is illustrated that M_k preserves its relative performance across all evaluated datasets. From the perspective of the similarity layer, as illustrated in Table. 4.2, it can be seen that there are actually two activation schemes that work best for each similar-

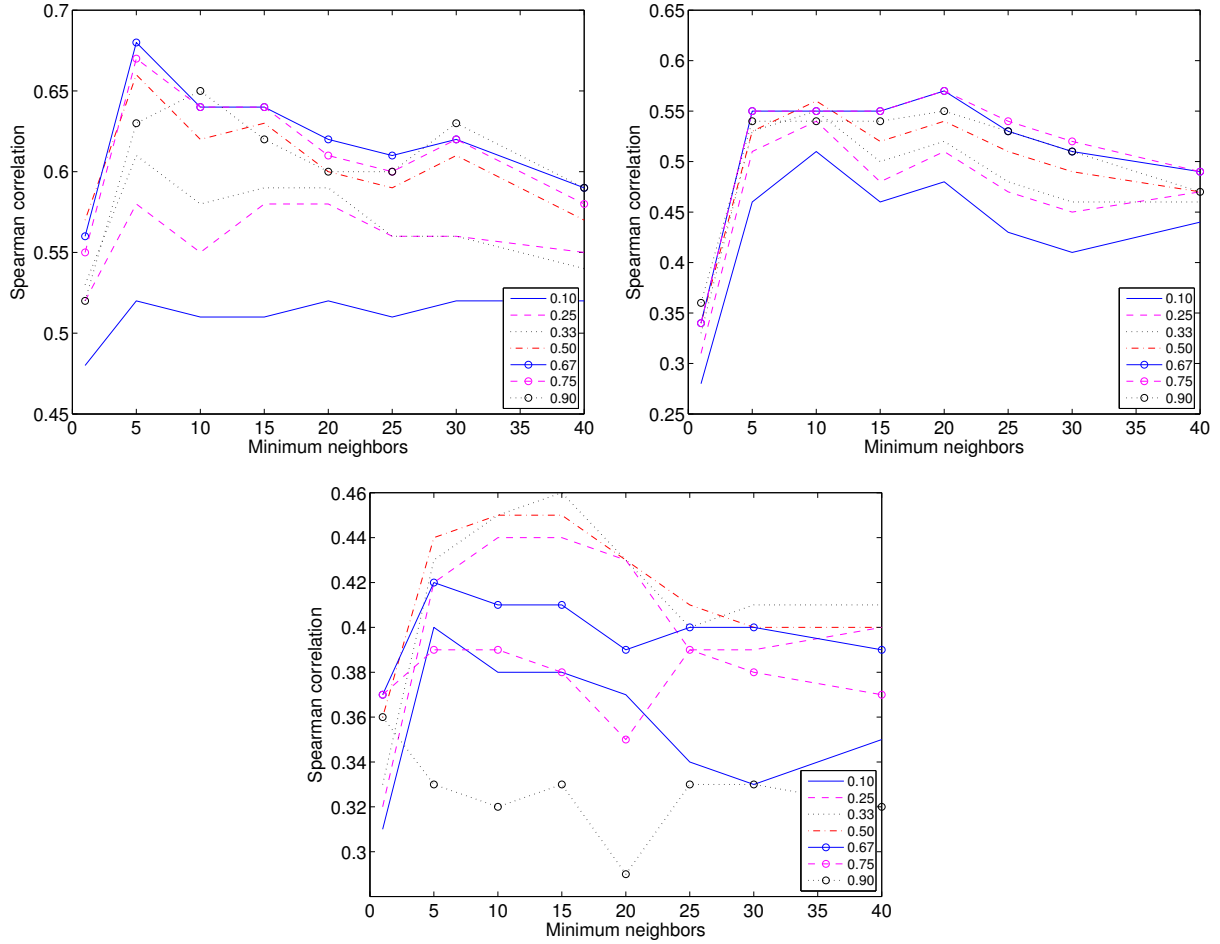

Figure 4.5: Performance of similarity asymmetry (M_2 metric) on a) noun-noun, b) adjective-noun, and c) verb-object datasets, using activation symmetry ($w_{act} = 0.5$) and the activation schemes described in Table 4.2.

ity metric, across all datasets, specifically, the *mostsimilar* scheme on the NN evaluation dataset, and the $inter_{var}$ scheme on the AN and VO datasets, with the exception of $inter_{var}$, for the case of the H metric on NNs, and *union*, for the case of P_k on ANs. In Fig. 4.1 it is observed that, although the $inter_{var}$ scheme is more robust across the three datasets and performs best in the case of ANs and VOs, the *mostsimilar* scheme over-performs it for the case of NNs. When considering the models as asymmetrical, performances can be boosted. However, it should be noted that asymmetry may be utilized better on a per-phrase basis, instead of an a priori setting. Smoothing the similarity metric range, for the case of M_k and P_k metrics, can also result in better performance.

In Table 4.2.3, evaluations are presented for the best model configurations, based on

their performance on the NN data set. In particular, we present model performances on NNs, ANs, and VOs, regarding the combinations that fit best the NN dataset, i.e., the selection of a) activation scheme, b) activation asymmetry (i.e., activation weight w_{act}), c) similarity metric, d) similarity smoothing (for the applicable metrics), and e) similarity asymmetry (i.e., similarity weight w_{sim}). In Table 4.2.3, the top NDSM performances are

Activation Layer		**Similarity Layer**		**NN**	**AN**	**VO**
Scheme	w_{act}	**Metric**	w_{sim}			
$inter_{fix}$	.50	M_1	.50	.56	.46	.37
$inter_{var}$	.50	M_1	.50	.58	**.57**	**.46**
$inter_{var}$	.50	M_*	.50	.62	**.57**	**.46**
$inter_{var}$	*	M_1	*	.58	**.57**	**.46**
$inter_{var}$	*	M_*	*	.65	**.57**	**.46**
headedge	.50	M_5	.50	.54	.47	.34
$inter_{var}$	.50	M_5	.50	.62	.52	.41
mostsimilar	.50	M_2	.50	.66	.46	.41
project	.50	M_5	.50	.46	.49	.41
union	.50	M_5	.50	.61	.45	.30
mostsimilar	.50	E	.50	.54	.21	.28
$inter_{var}$	.50	H	.50	.56	.39	.33
mostsimilar	.50	M_k	.50	.66	.49	.41
mostsimilar	.50	P_k	.50	.59	.45	.27
mostsimilar	.50	Q	.50	.63	.36	.31
headedge	.50	M_5	.75	.60	.48	.26
$inter_{var}$	.50	Q	.90	.66	.43	.36
mostsimilar	.67	Q	.50	**.71**	.42	.39
project	.90	M_5	.50	.54	.49	.40
union	.50	Q	.90	.62	.34	.30
mostsimilar	.67	E	.67	.59	.28	.40
mostsimilar	.67	H	.50	.62	.37	.33
mostsimilar	*	M_k	*	.70	.52	.43
mostsimilar	*	P_k	.50	.62	.48	.29
mostsimilar	.67	Q	.50	**.71**	.42	.39

Table 4.3: Model evaluations on NN, AN, and VO phrase pairs. Displaying models that perform best on NN evaluations, and categorized depending on performances against a) baseline models, b) activation schemes, c) similarity metrics, and d) model symmetry.

presented for the case of NNs, ANs, and VOs. Performances for the models described in [77] are presented alongside, for the purpose of comparing them with the proposed models. It is seen that using the variable- ($inter_{var}$) in place of the fixed-size intersection scheme

Eval. Set	Activation Layer		Similarity Layer		Score
	Scheme	Weight	Metric	Weight	
NN	*mostsimilar*	.67	Q	.50	.71
AN	$inter_{var}$	.50	M_1	.50	.57
VO	$inter_{var}$	.50	M_1	.50	.46

Table 4.4: Top overall model performances on NN, AN, and VO phrase pairs.

($inter_{fix}$), performance is boosted by an absolute 2%, 11% and 9% increase, for the case of NNs, ANs, and VOs, respectively, using the M metric. This boost in performance, when using the variable-size $inter_{var}$ scheme for compositional structures, is consistent with experimental observations from psycholinguistics [78], and shows that the activation area for phrases might be adaptive to the degree of relatedness between words. Performance is further boosted, for the case of NNs, by using the generalized M_k metric, or by using activation and/or similarity weighting, to reach an increase of 6% and 9%, respectively. The vast majority of the models perform better than the baseline for the case of NNs; however, their performance is not as stable for the case of ANs and VOs, and none of the additional proposed approaches (i.e., using alternate activation schemes, similarity metrics, or via asymmetry) achieve better performance than the asymmetric model of $inter_{var}$, combined with the M metric. Using asymmetry weighting generally improves the performance of the models. In general, the M_k scheme seems to work best with the proposed similarity metrics, when only considering symmetrical models; for asymmetrical models, however, the Q metric seems to gain ground. This can be attributed to the impact that asymmetry has in the attributional similarities computed by the Q metric, which provides it with more flexibility.

4.3 Fusion Models

We implemented three widely-used models from the literature to compare them with the proposed NDSM and fusion models, and in order to use them in order to form the latter. Specifically, we implemented the simple additive (*add*) and simple multiplicative (*mult*) models, proposed in [10, 11], as well as the lexical function (*lexfunc*) model, proposed in [73, 74]. These models are described in detail in Section 2.2.6. For the purposes of our experiments, and due to the strictly compositional approach of the *add* and *mult* models in composing semantics, as opposed to the transformational behavior of the *lexfunc* model,

we will refer to them in our experiments as compositional and transformational models, respectively. To realise the models, the DIStributional SEmantics Composition Toolkit (DISSECT [5], [87]) was used, which is part of the COMPOSES[6] (COMPositional Operations in SEmantic Space) project. The toolkit can be used for computing semantic spaces from co-occurrence matrices, while it integrates well known compositional functions to simplify the use of such models for similarity estimation between words or phrases. DISSECT also allows for the straightforward application of regression, dimensionality reduction, and other techniques that are typically used in DSMs.

In this section, we describe the experimental procedure that was followed for implementing and evaluating the literature and the fusion compositional models, proposed in Section 3.3. First, the approach to create the semantic space utilized by the models is presented, followed by the description of the selected parameters for realising them. Then, we display the evaluations of the models, along with the description of the results.

4.3.1 Model Configurations

Semantic space. In order to train *lexfunc*, a peripheral space was created to serve as the co-occurrence space for bigrams, since the model operates on observed bigram representations to learn the modifier weights. A peripheral space is a semantic space that extends the word (unigram) space with co-occurrence counts regarding observed bigram phrases, i.e., it encloses representations for bigrams that are computed in a holistic-based approach. To this end, in order to compose the peripheral space, co-occurrence counts were computed for modifier-noun bigrams. The co-occurrence matrix was composed after selecting all modifier-noun structures that occurred at least 50 times in the corpus, with the condition that the modifier occurrs in the evaluation datasets. For the input corpus, the same corpus upon which NDSMs were applied (described in 4.2.2) was used. Predefined, POS-tagged, content word lists were used for the selection of appropriate modifiers (nouns, adjectives, verbs), as well as for selecting the head nouns to compose the peripheral space. It was assumed that all nouns, adjectives, and verbs that precede a noun are considered as modifiers of the noun. The resulting space was re-weighted using PPMI (described in 2.2.5). Two dimensionality reduction techniques were utilized for the experiments, in particular a) SVD, and b) NMF (described in 2.2.2), resulting in two spaces that were further reduced to a) 300, and b) 500 dimensions. This produced four different semantic spaces,

[5] http://clic.cimec.unitn.it/composes/toolkit/

[6] http://clic.cimec.unitn.it/composes/

upon which we applied the models.

Transformational (*lexfunc*) model. In order to gather training data for the *lexfunc* model, we selected modifier-noun bigrams from the corpus comprising of a *modifier* that occurred in the test datasets, and used regression to learn the weights of the modifiers upon the peripheral space. To this end, two regression techniques from the literature were used, specifically a) Least Squares Regression (LSR), and b) Ridge Regression (RR). Both of these regression techniques are described in 2.2.3. This resulted in two variations of the model. We implemented the *lexfunc* models via the DISSECT toolkit, and applied them on all of the four semantic spaces.

Compositional (*add*, *mult*) models. In order to realise the *add* and *mult* models, the built-in functions of the DISSECT toolkit were utilized and applied on all four semantic spaces. We only experimented with the simple additive and simple multiplicative versions of the models.

Fusion (Φ^{lf}_{net}, Φ^{lf}_{add}) models. In order to implement the fusion models, proposed and described in 3.3, the best performing configurations for the component models were used. In particular, regarding Φ^{lf}_{net}, the models that performed best for the case of *lexfunc* and NDSMs were combined, by considering their performance on the NN dataset. To this end, we selected the appropriate compositional component as the NDSM that performed best for the case of NNs (presented in Table 4.2.3), while the *lexfunc* model on the NMF space of 300 dimensions and trained with RR was selected as the transformational component [7]. The same convention was followed for realising Φ^{lf}_{add}, i.e., the best performing *add* model on the NN dataset was used as the compositional component, specifically the model on the NMF space of 300 dimensions [8]. For this set of experiments, we set the weights $\alpha = 0.5$ and $\beta = 1$, as, by experimental tuning, we concluded that this set of weights fit best the model for the task.

4.3.2 Evaluation Results

Evaluation results for the literature *add* and *mult* (compositional), and the *lexfunc* (transformational) models, are presented in Table 4.3.2. For the *add* model, no significant changes in performance are observed between the spaces. For the case of the *mult* model, however, performance drops significantly when the SVD space is used. This makes sense, as

[7] The selected *lexfunc* model was the model that performed best on the NN dataset among the different *lexfunc* configurations that were evaluated, as is presented in Table 4.3.2.

[8] Also, the selected *add* model was the model that performed best on the NN dataset among the different *add* configurations that were evaluated, as is presented in Table 4.3.2.

Model				NN	AN	VO
Composition	**Dim. Reduction**	**Dimensions**	**Regression**			
add	NMF	300	-	**.67**	.61	.53
add	NMF	500	-	.66	**.63**	.56
add	SVD	300	-	.63	.59	**.59**
add	SVD	500	-	.66	**.63**	**.59**
mult	NMF	300	-	**.59**	**.38**	.36
mult	NMF	500	-	**.59**	.36	**.42**
mult	SVD	300	-	.36	.23	.21
mult	SVD	500	-	.36	.23	.21
lexfunc	NMF	300	RR	**.76**	**.46**	**.35**
lexfunc	NMF	300	LSR	.38	.24	.21
lexfunc	NMF	500	RR	.67	.41	.28
lexfunc	NMF	500	LSR	.30	.17	.06
lexfunc	SVD	300	RR	.63	.35	.26
lexfunc	SVD	300	LSR	.36	.25	.09
lexfunc	SVD	500	RR	.56	.33	.23
lexfunc	SVD	500	LSR	.36	.24	.10

Table 4.5: Performance of the simple additive (*add*), simple multiplicative (*mult*), and lexical function (*lexfunc*) models on NN, AN, and VO phrase pairs. Evaluations are reported using Spearman's correlation coefficient with human judgements.

the *mult* model is defined by pairwise multiplication and is expected to perform poorly when the composed vectors contain negative values, as is the case with SVD. Regarding the *lexfunc* model, a major impact in performance is caused when using RR for training, instead of LSR; RR has superior performance over LSR training in all configurations. This can be attributed to RR's better handling of the matrix multicollinearity problem, and the use of generalized cross-validation in order to compute the weight of λ (described in detail in 2.2.3). The *lexfunc* model, when using RR learning on the NMF space of 300 dimensions, performs best for the case of NNs, reaching an evaluation score of .76 for the case of NNs, which is the best among the literature models, It also achieves .46 and .35 for the case of ANs and VOs, respectively, which is the top performance among all *lexfunc* configurations. Best overall performances for ANs and VOs are obtained by the *add* model, with scores of .63 and .59, respectively. Next, evaluation results are presented for the Φ^{lf}_{net} and Φ^{lf}_{add} models in Table 4.3.2, along with the respective performances of the component models that compose them, i.e., *lexfunc* and *net*, for the Φ^{lf}_{net}, or *add*, for the Φ^{lf}_{add}. Detailed descriptions for the aforementioned models and their configurations are

Model	NN	AN	VO
lexfunc	**.76**	.46	.35
add	.67	**.61**	**.53**
fusion Φ^{lf}_{add}	**.76**	.60	.44
lexfunc	.76	.46	**.35**
net	.71	.37	.32
fusion Φ^{lf}_{net}	**.81**	**.51**	.33

Table 4.6: Performance of the fusion Φ^{lf}_{net} and Φ^{lf}_{add} models, along with their component models (*lexfunc*, *net*, and *add*) on NN, AN, and VO phrase pairs. Evaluations are reported using Spearman's correlation coefficient with human ratings.

provided in Section 4.3.1. Φ^{lf}_{add} yields no relative improvements over the best performances of the separate models. Specifically, for the case of NNs, Φ^{lf}_{add} merely achieves the same score with *lexfunc* (.76), while, for ANs and VOs, it fails to improve over its component *add* model. Φ^{lf}_{net} has an improved score, for the case of NNs, with a score of .81 which is also the best observed performance overall and provides an absolute increase of 5% when compared to the score of its component *lexfunc* model, which is the best performance of all models in isolation. Φ^{lf}_{net} also improves over its components' performances for the case of ANs, reaching a score of .51, which is an improvement over the scores of .46 and .37 of *lexfunc* and *net*, respectively. Φ^{lf}_{net} also fails to improve performance over its components, for the case of VOs, lying in-between the performances of its components, however such differences reside within the range of statistical insignificance.

4.4 Summary

In this chapter, we presented the experimental setup and evaluations of the proposed network-based and fusion models described in Chapter 3. The use of a variable-size $inter_{var}$ activation scheme improved performance over the baseline $inter_{fix}$ for the case of NNs at an absolute 2%, 11%, and 9% for the case of NNs, ANs, and VOs, respectively, proposing that an adaptable scheme can better handle semantics for complex phrases when such meanings are encoded as a network graph. The *mostsimilar* scheme performed best for the case of NNs. The ability of the scheme to selectively compute the composed neighborhood renders it able to be used for structures of arbitrary length without scalability issues. We investigated the use of k as a smoothing factor for M_k and showed that it also has a significant effect on the performance of the models. We evaluated model asymmetry,

showing that performance can be boosted when appropriately shifting the contribution degree of each phrase constituent. Combining merely compositional with transformational models under a fusion scheme can have positive effects on performance. This has been underlined by the top performance of .81 for the case of NNs, when combining *lexfunc* with the network-based models. The latter model also improved over the constituent models' performances for the case of ANs. The aforementioned suggest that a fusion scheme that combines different strategies for estimating compositional semantics is a sensible approach for handling the different functions encoded within complex structures.

Chapter 5

Discussion

In the previous chapters, we described strategies for semantic composition and similarity estimation via the utilization of extrinsic, at word level, or intrinsic, at phrase level, context. Specifically regarding the phrase level, various approaches from the literature were presented for deriving semantics for complex structures via the semantic composition of the meanings their parts. Next, a variety of models were proposed, based on the utilization of semantic neighbors, that handle semantics through the notion of activational priming. Specifically, we presented activation-based techniques for composing such areas as a function of the neighborhoods of the (complex) structures' constituents, as well as various metrics, motivated by psycholinguistics and metric space algebra, that estimate similarity by utilizing these activations. We also presented a novel approach that exploits the transformational properties of phrases, as those are defined by their modifiers, in order to reduce similarity estimation to a fusion metric that combines different models that estimate semantics from different perspectives. Finally, we evaluated some of the most widely used models along with those presented in this thesis, i.e., the proposed activation schemes and similarity metrics, as well as the fusion model. Results of the evaluation of the fusion models, along with the respective performances of their component models, were displayed in Table 4.3.2. The performance of the fusion model provides an insight concerning the variability that exists regarding the established relational type between the constituents that form a complex linguistic expression; compositional semantics manifest themselves in a variety of ways that is based on their constituents and the fashion in which these constituents affect the meaning of the whole. Related work on bigram structures [8, 73, 85, 86] has shown that the meaning of such a structure can be affected by the functional influence of its modifier, with respect to the other components, while other structures have proven to be semantically represented more accurately via a direct composition of the meanings of their parts. Words that act as *functions* on their linguistic context have attracted much interest, and have successfully been handled by computational models. It has been shown [88] that such modifications can be successfully modeled by distributional semantics and that, at least for some cases, where the modifier has strong transformational

properties, higher-order predication fares better than a simple composition of senses. It has also been shown that modifiers are activated by specific properties of their immediate preceding context. In Section 3.3.1, we took a closer look on modifiers, defined a measure that estimates their transformative degree, and attempted to detect the types of phrases that express high transformational properties, as a means to typify their behaviour and prerequisites, when using them to derive semantics or to estimate similarities. We believe that, through the detection of such properties on the structure modifier, a decision can be deducted with respect to the nature of the compositionality type that best describes said structure, i.e., whether the phrase has mainly a transformational or a strictly compositional effect. For example, consider the phrases "normal cat", and "dead cat": the modifier "normal" has a much less transformational effect on the meaning of "cat" than the modifier "dead" has. It can be inferred that, since structure semantics depend on the semantic properties that are activated via the interaction between the components, modifiers can exhibit different functional behaviors, based on their immediate context, i.e., a modifier can affect the meaning of its encompassing phrase in varying degrees, based on its other constituents. Notwithstanding, the modifier itself plays an important role in modifying a structure's meaning. Another point to consider it that the effect of the modifier may vary, depending on the modifier's grammatical category, i.e., whether the modifier is, e.g., a verb, an adjective or a noun. However, these grammatical types are also further categorized: for example, nouns can be abstract or concrete, adjectives can be intensional or not, and verbs can be transitive or intransitive [88, 89]. The appropriate compositional method that will form and utilize the semantics for such complex structures should be adaptive to the transformational degree residing within the phrase, as well as to the functional behaviour of the modifier.

5.1 NDSM Application on Longer Structures

One of the complications related to a large portion of the compositional models, proposed in the literature, is their lack of flexibility, when considering them for estimating semantics of longer structures; scaling them to n-grams of increased word length is inversely proportional to their utilization. In Chapter 3, we described compositional NDSMs, as adapted to the use case of bigrams. In this section, we investigate the application of NDSMs, as described in this thesis, to the case of structures of arbitrary length, as are sentences. To this end, an experiment was realised in order to observe the performance of the proposed NDSM to the

case of said structures. To evaluate the experiment, we used the SICK (Sentences Involving Compositional Knowledge) dataset [90], which has been selected as the standard evaluation dataset for Semeval 2014 Task 1 [1] and other work that is related with this task. The dataset consists of 10,000 English sentence pairs, containing rich lexical, syntactic and semantic aspects that compositional DSMs should by able to detect, while it ignores sentential phenomena that are not within the scope of compositional distributional semantics, such as idiomatic multi-word expressions, named entities, etc. Each pair is annotated a score regarding the semantic relatedness and the entailment relation between them; we only considered semantic relatedness for the experiment. As the semantic network, we used the semantic network described in Section 4.2.1, and used for the experiments on NDSMs. Once more, we evaluated the experiments using Spearman's correlation coefficient, defined in Eq. 4.1. To this end, the *mostsimilar* scheme was utilized for the activation layer, based on its ability to scale to structures of arbitrary length. For the similarity layer, the H metric was selected, due to its better relative performance with respect to the other metrics [2]. We only experimented with the symmetrical version of the model, i.e., no weighting biases were considered at the activation or the similarity layer (i.e., $w_{act} = w_{sim} = 0.5$).

The model achieved a .27 score when correlated with human judgements. Although being a poor performance, it indicates that the model can be applied to sentences of arbitrary length and provides a reference for future experiments. Moreover, it should be mentioned that much of the configuration used for this simple experiment does not adhere with the linguistic, syntactic, and semantic properties of such complex structures. Based on predefined configurations, the model encodes in the same way every possible relation between the constituents of a sentence, using a fixed activation and similarity layer approach, which is not a sensible approach for the case of sentential structures. The various activation schemes proposed in Section 3.2.1 serve as functions that describe a variety of relational phenomena between the constituents that are encoded within a complex structure and are based on syntactic, semantic and grammatical constraints. We intuitively believe that semantics for structures of arbitrary length can be computed by forming a general NDSM model that utilizes its compositional schemes under a function that considers the relational types between the structure's constituents, and appropriately select the most fit scheme (e.g., through a supervised learning way). The semantics could

[1] http://alt.qcri.org/semeval2014/task1/

[2] We also experimented with other similarity metrics, such as the M_k and the Q metrics, but those metrics failed to produce any mentionable performance. A more conclusive evaluation of the models could be realised in future work that could investigate the relative performance between the models.

then be formed by a recursive bottom-up procedure that considers increasingly convoluted structures. This approach has also been adopted by other work related with the estimation of such semantics [50, 75].

5.2 Conclusions

In this thesis, a network-based approach was presented that operates on neighborhoods of variable size in order to compose semantics and model similarity of compositional structures. We investigated and presented five activation schemes, motivated by semantic priming, for composing activation areas for complex structures, and four similarity metrics, motivated by psycholinguistics and metric space algebra, for estimating similarity via the utilization of those areas. It was shown that, by employing variable size activation, performance for the task of semantic similarity can be boosted, and each bigram structure can exhibit different behaviour that should be handled respectively, in order to successfully imprint it in activations, i.e., each structure can be assigned to a specific type of compositional function. The fusion model state-of-the-art performance of 81% Spearman correlation with human judgements, when combining the proposed NDSM with the *lexfunc* model for the case of NN, suggests that combining different means of semantic models is a sensible way to measure the contribution of different similarity metrics in order to adapt similarity computation to the compositional type for each structure. Using the modifier's transformative degree can serve to quantify the transformational properties of the structure and, thus, contribute to the utilization of its semantics for the task of semantic similarity. The fusion scheme failed to improve on the case of VOs, however, for the case of ANs, it achieved improved performance when network-based models were used. This can be attributed to the behavior of verbs and adjectives in a phrase: verbs have a fully functional effect on the constituents, while adjectives base their behavior on context. We leave it to future work to further investigate how compositional semantics for such syntactic types may be handled more appropriately using these schemes.

Investigation of the modifiers, after ranking them according to their transformative degree, provided an insight to the modifier types that associate with a high or low transformative degree and, subsequently, their role as mostly *compositional* or *transformational*. Our observations suggest that modifiers affect the structure in which they are observed in different ways. Some modifiers have a stronger effect on the meaning of the head word, while others act merely as constituents of simple compositions. The proposed fusion of

the transformational, *lexfunc* model, with NDSMs or simple compositional models, as is described in Section 3.3, indicates that combining different models can yield improved performance when the transformative degree of modifiers is used as a fusion criterion.

5.3 Contributions and Future Directions

An activation-based approach of modeling compositional semantics was presented by using cognitively-motivated ideas from psycholinguistics and semantic priming. The availability of different compositional schemes for computing activations results in a model that can be flexible to the intrinsic relations between the constituents for each phrase, and can thus be used to adaptively model semantics for said structures. Our proposal of different similarity metrics introduces a variety of approaches that depend on well-founded theories from psycholinguistics and metric space algebra, in order to model the task of semantic similarity by reducing it to comparisons between metric spaces. The current thesis also presents a measure that defines the transformative degree as an indicator of the transformational properties of a phrase modifier, as well as a fusion model that can regulate the contribution of a transformational and a compositional model for similarity estimation, based on said measure. Compositional structures behave in a variety of ways, based on the composition itself and the relational types among their constituents, and the transformational degree serves as a fit criterion for the compositional type of the phrase and for deciding on the approach of utilizing its semantics.

We aim at building on these contributions in order to improve the models proposed in this thesis. In future work, the role of modifiers can be further investigated, as well as their utilization in the presented activation composition approaches. The criteria for deriving and choosing activations is also a matter of further research. In the proposed schemes, we only experimented with activations of the same size between the constituents, while constituents could merely utilize sub-areas of their neighborhoods, based on their adjacent intrinsic context. Moreover, the application of NDSMs on longer semantic structures needs to be further investigated. In Section 5.1 we presented a very basic strategy for porting the proposed models to estimating semantics for such structures. Although the performance of the sentence-level model is poor (.27), we believe that this first attempt could serve as a stepping stone for further experiments. Strong improvements could be achieved by exploiting the flexibility offered by the various proposed schemes and metrics to address the various functions encountered withing a sentence. For example, the activation schemes

and similarity metrics proposed in this thesis could be utilized for computing activations and similarity on structures of arbitrary length and for detecting the variability of the relational types that exist within these structures. This could be realised by combining them in a way that considers the various lexical, syntactic and semantic phenomena that emerge at that level. The proposed fusion model can also be further enhanced and further investigated. Ideas from machine learning could be integrated for training the involved parameters in a supervised way. For example, α and β are predefined parameters in the proposed fusion model; training the weights using a training set would be sensible, in order find the optimal values. The smoothing function that acts on the transformative degree T can also be replaced, by experimenting with others suggested from the literature (e.g., log, tanh, exp, etc.). Finally, modifier behavior should be further investigated, especially when considering different types of associated head words; modifiers are expected to behave differently, based on the head word that precedes them, and such behavior should be modeled in order to accurately estimate the composed meaning of the structure.

Bibliography

[1] I. Androutsopoulos and P. Malakasiotis, "A survey of paraphrasing and textual entailment methods," *Journal of Artificial Intelligence Research*, vol. 38, pp. 135–187, 2010.

[2] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Distributional semantic models for affective text analysis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 11, pp. 2379–2392, 2013.

[3] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[4] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and cognitive processes*, vol. 6, no. 1, pp. 1–28, 1991.

[5] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, no. 1, pp. 141–188, 2010.

[6] M. Baroni and A. Lenci, "Distributional memory: A general framework for corpus-based semantics," *Computational Linguistics*, vol. 36, no. 4, pp. 673–721, 2010.

[7] M. Sahlgren, "The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces," Ph.D. dissertation, Department of Linguistics, Stockholm University, 2006.

[8] B. Coecke, M. Sadrzadeh, and S. Clark, "Mathematical foundations for a compositional distributional model of meaning," *Linguistic Analysis*, vol. 36, pp. 345–384, 2011.

[9] F. J. Pelletier, "The principle of semantic compositionality," *Topoi*, vol. 13, no. 1, pp. 11–24, 1994.

[10] J. Mitchell and M. Lapata, "Vector-based models of semantic composition." in *Proc. of ACL*, 2008, pp. 236–244.

[11] ——, "Composition in distributional models of semantics," *Cognitive science*, vol. 34, no. 8, pp. 1388–1429, 2010.

[12] T. Polajnar, L. Rimell, and S. Clark, "Evaluation of simple distributional compositional operations on longer texts," in *Proc. of LREC*, 2014.

[13] S. Clark and S. Pulman, "Combining symbolic and distributional models of meaning." in *AAAI Spring Symposium: Quantum Interaction*, 2007, pp. 52–55.

[14] D. Widdows, "Semantic vector products: Some initial investigations," *Second AAAI Symposium on Quantum Interaction*, vol. 26, p. 28th, 2008.

[15] S. Clark, "Vector space models of lexical meaning," *Handbook of Contemporary Semantics, Wiley-Blackwell, à paraître*, 2012.

[16] E. Iosif and A. Potamianos, "Similarity computation using semantic networks created from web-harvested data," *Natural Language Engineering*, vol. 21, no. 01, pp. 49–79, 2015.

[17] S. Georgiladakis, E. Iosif, and A. Potamianos, "Fusion of compositional network-based and lexical function distributional semantic models," *Proc. of CMCL*, pp. 39–47, 2015.

[18] T. P. McNamara, *Semantic priming: Perspectives from memory and word recognition*. Psychology Press, 2005.

[19] K. Mahesh and S. Nirenburg, *Knowledge-Based Systems for Natural Language Processing*. Computing Research Laboratory, New Mexico State University, 1996.

[20] P. Roget, *Roget's Thesaurus of English Words and Phrases*. Longman Group Ltd., 1852.

[21] M. Jarmasz and S. Szpakowicz, "Roget's thesaurus and semantic similarity1," *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, vol. 2003, p. 111, 2004.

[22] C. Fellbaum, *WordNet*. Wiley Online Library, 1998.

[23] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis." in *Proc. of IJCAI*, vol. 7, 2007, pp. 1606–1611.

[24] A. Budanitsky and G. Hirst, "Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures," in *Workshop on WordNet and Other Lexical Resources*, 2001.

[25] ——, "Evaluating WordNet-based measures of semantic distance," *Computational Linguistics*, vol. 32, pp. 13–47, 2006.

[26] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *Proc. of IJCAI*, vol. 3, 2003, pp. 805–810.

[27] P. Resnik *et al.*, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *J. Artif. Intell. Res.(JAIR)*, vol. 11, pp. 95–130, 1999.

[28] D. Lin, "An information-theoretic definition of similarity." in *Proc. of ICML*, vol. 98, 1998, pp. 296–304.

[29] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *ROCLING'97*, 1997.

[30] G. Grefenstette, *Explorations in Automatic Thesaurus Discovery*. Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[31] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proc. of the 5th annual international conference on Systems documentation*. ACM, 1986, pp. 24–26.

[32] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet: An electronic lexical database*, vol. 305, pp. 305–332, 1998.

[33] M. Sussna, "Word sense disambiguation for free-text indexing using a massive semantic network," in *Proc. of the second international conference on Information and knowledge management*. ACM, 1993, pp. 67–74.

[34] M. J. Sussna, "Text retrieval using inference in semantic metanetworks," Ph.D. dissertation, University of California, San Diego, 1997.

[35] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proc. of ACL*. Association for Computational Linguistics, 1994, pp. 133–138.

[36] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.

[37] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proc. of IJCAI*, 1995, pp. 448–453.

[38] S. P. Ponzetto and M. Strube, "Knowledge derived from wikipedia for computing semantic relatedness." *J. Artif. Intell. Res.(JAIR)*, vol. 30, pp. 181–212, 2007.

[39] M. Strube and S. P. Ponzetto, "Wikirelate! computing semantic relatedness using wikipedia," in *Proc. of AAAI*, 2006, pp. 1419–1424.

[40] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proc. of AAAI*, vol. 6, 2006, pp. 775–780.

[41] G. Grefenstette, *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, 1994.

[42] P. D. Turney, "Similarity of semantic relations," *Computational Linguistics*, vol. 32, no. 3, pp. 379–416, 2006.

[43] S. Padó and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, 2007.

[44] K. Erk and S. Padó, "A structured vector space model for word meaning in context," in *Proc. of EMNLP*. Association for Computational Linguistics, 2008, pp. 897–906.

[45] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JAsIs*, vol. 41, no. 6, pp. 391–407, 1990.

[46] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[47] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[48] T. Hastie, R. Tibshirani, and J. Friedman, *Unsupervised Learning*. Springer New York, 2009, vol. 2.

[49] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[50] E. Grefenstette, G. Dinu, Y.-Z. Zhang, M. Sadrzadeh, and M. Baroni, "Multi-step regression learning for compositional distributional semantics," *Proc. of IWCS*, 2013.

[51] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[52] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proc. of ACL*. Association for Computational Linguistics, 2012, pp. 873–882.

[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ICLR Workshop*, 2013.

[54] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. of ACL*, vol. 1, 2014, pp. 238–247.

[55] P. Vitanyi, "Universal similarity," in *Proc. of Information Theory Workshop on Coding and Complexity*, 2005, pp. 238–243.

[56] R. L. Cilibrasi and P. M. Vitanyi, "The Google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370–383, 2007.

[57] J. Gracia, R. Trillo, M. Espinoza, and E. Mena, "Querying the web: A multiontology disambiguation method," in *Proc. of ICWE*, 2006, pp. 241–248.

[58] S. Bordag, "Elements of knowledge-free and unsupervised lexical acquisition," Ph.D. dissertation, Univ. Leipzig, 2007.

[59] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behavior research methods*, vol. 39, no. 3, pp. 510–526, 2007.

[60] A. Pargellis, E. Fosler-Lussier, C.-H. Lee, A. Potamianos, and A. Tsai, "Auto-induced semantic classes," *Speech Communication*, vol. 43, no. 3, pp. 183–203, 2004.

[61] A. Pargellis, E. Fosler-Lussier, A. Potamianos, and C.-H. Lee, "A comparison of four metrics for auto-inducing semantic classes," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on.* IEEE, 2001, pp. 218–221.

[62] L. Lee, "Similarity-based approaches to natural language processing," Ph.D. dissertation, Harvard University, 1997.

[63] ——, "Measures of distributional similarity," in *Proc. of ACL.* Association for Computational Linguistics, 1999, pp. 25–32.

[64] G. Lapesa and S. Evert, "A large scale evaluation of distributional semantic models: Parameters, interactions and model selection," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 531–545, 2014.

[65] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd," *Behavior research methods*, vol. 44, no. 3, pp. 890–907, 2012.

[66] P. D. Turney, "Domain and function: A dual-space model of semantic relations and compositions," *Journal of Artificial Intelligence Research(JAIR)*, vol. 44, pp. 533–585, 2012.

[67] R. Montague, "Deterministic theories," *Formal Philosophy*, 1974.

[68] G. Lakoff, "Linguistic gestalts," in *Papers from the... Regional Meeting. Chicago Ling. Soc. Chicago, Ill.*, vol. 13, 1977, pp. 236–287.

[69] G. Frege and J. L. Austin, *The Foundations of Arithmetic: A logico-mathematical enquiry into the concept of number.* Northwestern University Press, 1980.

[70] G. Nunberg, I. A. Sag, and T. Wasow, "Idioms," *Language*, pp. 491–538, 1994.

[71] W. Kintsch, "Predication," *Cognitive Science*, vol. 25, no. 2, pp. 173–202, 2001.

[72] S. Clark, B. Coecke, and M. Sadrzadeh, "A compositional distributional model of meaning," in *Proc. of the Second Quantum Interaction Symposium (QI-2008)*, 2008, pp. 133–140.

[73] M. Baroni and R. Zamparelli, "Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space," in *Proc. of EMNLP.* Association for Computational Linguistics, 2010, pp. 1183–1193.

[74] M. Baroni, R. Bernardi, and R. Zamparelli, "Frege in space: A program of compositional distributional semantics," *Linguistic Issues in Language Technology (LiLT)*, vol. 9, 2014.

[75] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. of the 2012 Joint Conference on EMNLP and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1201–1211.

[76] F. M. Zanzotto, I. Korkontzelos, F. Fallucchi, and S. Manandhar, "Estimating linear models for compositional distributional semantics," in *Proc. of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1263–1271.

[77] E. Iosif, "Network-based distributional semantic models," Ph.D. dissertation, School of Electronic and Computer Engineering, Technical University of Crete, Kounoupidiana, Chania, 2013.

[78] D. N. Osherson and E. E. Smith, "On the adequacy of prototype theory as a theory of concepts," *Cognition*, vol. 9, no. 1, pp. 35–58, 1981.

[79] G. Athanasopoulou, E. Iosif, and A. Potamianos, "Low-dimensional manifold distributional semantic models," *Proc. of COLING, Dublin, Ireland*, 2014.

[80] P. Gärdenfors, *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[81] J. Karlgren, A. Holst, and M. Sahlgren, "Filaments of meaning in word space," in *Advances in Information Retrieval*. Springer, 2008, pp. 531–538.

[82] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman, "How to grow a mind: Statistics, structure, and abstraction," *science*, vol. 331, no. 6022, pp. 1279–1285, 2011.

[83] P. Resnik, "Using information content to evaluate semantic similarity in a taxanomy," in *Proc. of International Joint Conference for Artificial Intelligence*, 1995, pp. 448–453.

[84] W. L. Hung and M. S. Yang, "Similarity measures of intuitionistic fuzzy sets based on hausdorff distance," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1603–1611, 2004.

[85] G. Boleda, M. Baroni, N. T. Pham, and L. McNally, "Intensionality was only alleged: On adjective-noun composition in distributional semantics," in *Proc. of IWCS 2013*, 2013, pp. 35–46.

[86] G. Kruszewski and M. Baroni, "Dead parrots make bad pets: Exploring modifier effects in noun phrases," *Lexical and Computational Semantics (* SEM 2014)*, p. 171, 2014.

[87] G. Dinu, N. T. Pham, and M. Baroni, "DISSECT-DIStributional SEmantics Composition Toolkit," in *Proc. of ACL: System Demonstrations*, 2013, pp. 31–36.

[88] G. Boleda, M. Baroni, L. McNally, and N. Pham, "Intensionality was only alleged: On adjective-noun composition in distributional semantics," in *Proc. of IWCS*, 2013, pp. 35–46.

[89] N. Malandrakis and S. Narayanan, "Therapy language analysis using automatically generated psycholinguistic norms," in *Proc. of Interspeech*, 2015.

[90] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A sick cure for the evaluation of compositional distributional semantic models," in *Proc. of LREC*. Citeseer, 2014, pp. 216–223.