TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRONIC AND COMPUTER ENGINEERING
TELECOMMUNICATIONS DIVISION

Semantic Similarity Computation and Word Sense Induction using Hidden Sets Multidimensional Scaling

by

Georgia Athanasopoulou

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

ELECTRONIC AND COMPUTER ENGINEERING

July 9, 2016

THESIS COMMITTEE

Associate Professor Alexandros Potamianos, *Thesis Supervisor*
Associate Professor Polychronis Koutsakis
Professor Athanasios Liavas

Abstract

In this thesis, motivated by evidences in psycholinguistics and cognition, we propose an unsupervised language-agnostic Distributional Semantic Model (DSM), that utilize web harvested data, for the problem of semantic similarity estimation. Semantic similarity can be applied to numerous tasks of Natural Language Processing (NLP), such as affective text analysis and paraphrasing. In the first part of the thesis, the construction of typical DSMs following the well-established Vector Space Model, is presented. More specifically, we describe the creation of corpora by harvesting web documents following a query-based approach, as well as state-of-the-art DSMs used for the computation of semantic similarity from the corpora. Next, we propose a novel hierarchical distributed semantic model (DSM), that is inspired by evidence in psycholinguistics and cognition, and consists of low-dimensional manifolds built on semantic neighborhoods. Each manifold is sparsely encoded and mapped into a low-dimensional space. Global operations are decomposed into local operations in multiple sub-spaces; results from these local operations are fused to come up with semantic relatedness estimates. Manifold DSM are constructed starting from a pairwise word-level semantic similarity matrix. The proposed model is evaluated against state-of-the-art/baseline DSMs on semantic similarity estimation task, where the similarity metrics are evaluated against human similarity ratings. The proposed model significantly improve performance comparing to the baseline approaches for the task of semantic similarity estimation between words. Furthermore the proposed model is evaluated in a taxonomy task achieving achieving state-of-the-art results. Finally, motivated by evidence of cognitive organization of concepts based on the degree of concreteness, we present the performance of proposed DSM for abstract and concrete nouns.

Περίληψη

Σε αυτή την μεταπτυχιακή εργασία, εμπνευσμένοι από στοιχεία της ψυχογλωσσολογίας και τη γνωσιακής επιστήμης, προτείνουμε την κατασκευή ενός κατανεμημένου σημασιολογικού μοντέλου (Distributional Semantic Model - DSM) το οποίο μπορεί να εφαρμοστεί με επιτυχία σε διάφορες γλώσσες και λειτουργεί χωρίς επίβλεψη χρησιμοποιώντας δεδομένα που συλλέγονται το ίντερνετ. Αυτό το μοντέλο το εφαρμόσαμε στο πρόβλημα της εκτίμησης της σημασιολογικής ομοιότητας μεταξύ λέξεων. Η σημασιολογική ομοιότητα είναι μία μετρική η οποία μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές του τομέα της Επεξεργασίας Φυσικής Γλώσσας, (Natural Language Processing (NLP) όπως η συναισθηματική ανάλυση κειμένου και παράφραση κειμένων. Στο πρώτο μέρος της διατριβής, παρουσιάζεται η κατασκευή γνωστών σημασιολογικών μοντέλων της βιβλιογραφίας τα οποία κάνουν χρήση διανυσμάτων, Vector Space Model (VSM). Πιο συγκεκριμένα, αρχικά περιγράφουμε τον τρόπο δημιουργίας μεγάλων αρχείων κειμένου από τη συγκέντρωση εγγράφων από το ίντερνετ και μετέπειτα παρουσιάζουμε συστήματα DSMs της βιβλιογραφίας που χρησιμοποιούνται για την υπολογισμό της σημασιολογικής ομοιότητας με βάση τα αρχεία κειμένου. Στη συνέχεια της διατριβής, προτείνουμε ένα νέο ιεραρχικό κατανεμημένο σημασιολογικό μοντέλο (DSM), το οποίο είναι εμπνευσμένο από στοιχεία της ψυχογλωσσολογίας και τη γνωστική επιστήμης, και αποτελείται από παράλληλες χαμηλών διαστάσεων σημασιολογικές γειτονιές. Σε κάθε γειτονιά αντιστοιχίζεται μία αραιή κωδικοποίηση και έπειτα η γειτονία προβάλλεται σε ένα χαμηλών διαστάσεων υποχώρο. Έπειτα ο υπολογισμός της σημασιολογικής ομοιότητας μεταξύ λέξεων, αποσυντίθεται αρχικά σε τοπικές πράξεις πάνω στους πολλαπλούς παράλληλους υποχώρους και τελικά τα αποτελέσματα αυτών συνθέτονται για να καταλήξουμε σε μία γενική απόφαση. Το προτεινόμενο μοντέλο αξιολογείται έναντι γνωστών αλγορίθμων της βιβλιογραφίας στην εκτίμηση της σημασιολογικής της ομοιότητας σύνολων από ζευγάρια λέξεων τα οποία έχουν βαθμολογήσει άνθρωποι για την σημασιολογική τους ομοιότητα. Το προτεινόμενο μοντέλο έχει βελτιώσει σημαντικά τις επιδόσεις σε σύγκριση με άλλα μοντέλα της βιβλιογραφίας που λειτουργούν χωρίς επίβλεψη στην εκτίμηση της σημασιολογικής ομοιότητας μεταξύ των λέξεων.

Acknowledgements

First of all I would like to thank my committee, Alexandros Potamianos, Polychronis Koutsakis and Athanasios Liavas for honoring me with their participation to my committee and for supporting me in this effort.

I would like to express my deepest thanks to my mentor Alexandros Potamianos for his immeasurable support and for all the things that he has taught me. I acknowledge his effort to evolve me as a professional, he has changed the way I work, the way I think, the way I address difficult situations, my 'decision system' and generally me as a person.

I would also like to thank my research group for all the process of sharing ideas and teamwork.

Special thanks to my family for their continuous support at any possible level and for giving me motivation to continue my studies.

Additionally, I would like to thank my best friends Vicky Prokopi, Elisavet Palogiannidi and Katerina Malisova for supporting me and for the carefree moments that I have lived and still living with them.

Last but not least, I would like to deeply thank Panos Alevizos because he supports me with any possible way, he provides me an alternative way of thinking, he believes in me, he encourages me and because he has made me a better person.

To Alexandra Gardikioti

Contents

List of Figures

List of Abbreviations

DSMs	Distributional Semantic Models
VSM	Vector Space Model
SDS	Spoken Dialoue Systems
MDS	Multidimensional Scaling
PCA	Principal Component Analysis
LLE	Local Linear Embedding
SP	Sparse Projection

Notation

x	a variable
$\mathbf{x}$	a vector
$\mathbf{x}(i)$	the i^{th} item of $\mathbf{x}$ vector
$\mathbf{A}$	a matrix
$\mathbf{A}(i)$	the i^{th} row of $\mathbf{A}$ matrix
$\mathbf{A}(i,j)$	the j^{th} item of i^{th} row of $\mathbf{A}$ matrix
$\mathbf{A}^\top$	transpose of matrix $\mathbf{A}$
$\mathbf{I}_n$	$n \times n$ identity matrix
$\lvert\mathcal{C}\rvert$	the cardinality of a set $\mathcal{C}$
$\lvert\lvert\mathbf{x}\rvert\rvert_p$	the p norm of a vector $\mathbf{x}$
$\mathbb{R}$	the set of real numbers
$\mathbb{B}$	the set of binary numbers
$\arg\max_{\mathbf{x}}\{f(\mathbf{x})\}$	the argument that maximizes $f(\mathbf{x})$

Chapter 1

Introduction

1.1 Representation of Lexical Semantics

From cognitive experiments one can easily deduce that there exist fundamental cognitive relationships between terms/words [2, 3]. Semantic similarity relationship corresponds to the degree of likeness of meaning or semantic content between two terms and is a fundamental relationship of human cognition [4]. Word association is a more low-level fundamental relationship between terms and expresses the degree of semantic or pragmatic relatedness, for example, 'leaf' is similar to 'flower', but is only related to 'spring' and 'autumn' [5, 6]. Antonyms, e.g. 'wealth' and 'poverty', constitute an example of words with low semantic similarity and high association. Another example of words with low semantic similarity and high association, is the words related by cause and effect, e.g. 'gun' and 'kill'. The main features that humans employ to acquire associations are the co-occurrence and proximity [5, 6], namely these features, from all of our senses, e.g. from language, images, music, smells, scripts etc, are behind the cognitive acquisition and organization of our knowledge. Another important feature, that can be used in spoken language to extract associations and semantic relations, additively to co-occurrence and proximity, is context similarity. The motivation behind contextual similarity is that the similarity of context implies the similarity of meaning, so for the computation of context similarity between terms/words, the contexts of the words are utilized and thus context similarity captures both syntactic and pragmatic relations between words. Language engineers have modeled such associations by using features such as co-occurrence and proximity creating n-gram models of language [7].

The estimation of semantic similarity between words, sentences and documents is a fundamental problem for many research fields including computational linguistics [8], semantic web [9], cognitive science and artificial intelligence [10, 11]. In this work, we study the geometrical structure of the lexical space in order to extract semantic relations among words. In [12], the high-dimensional lexical space is assumed to consist of manifolds of very low dimensionality that are embedded in this high dimensional space. The manifold

hypothesis is compatible with evidence from psycholinguistics and cognitive science. In [13], the question *"How does the mind work?"* is answered as follows: cognitive organization is based on domains with similar items connected to each other and lexical information is represented hierarchically, i.e., a domain that consists of similar lexical entries may be represented by a more abstract concept. An example of such a domain is {*blue, red, yellow, pink, ...*} that corresponds by the concept of *color*. An inspiring analysis about the geometry of thought, as well as cognitive evidence for the low-dimensional manifold assumption can be found in [14], e.g., the domain of color is argued to be cognitively represented as an one-dimensional manifold.

1.2 Thesis Contribution

There has been much research interest on devising data-driven approaches for estimating semantic similarity between words. DSMs [15] are based on the distributional hypothesis of meaning [16] assuming that semantic similarity between words is a function of the overlap of their linguistic contexts. DSMs are typically constructed from co-occurrence statistics of word tuples that are extracted on existing corpora or on corpora specifically harvested from the web.[1] In [17], general-purpose, language-agnostic algorithms were proposed for estimating semantic similarity using no linguistic resources other than a corpus created via web queries. The key idea of this work was the construction of semantic networks and semantic neighborhoods that capture smooth co-occurrence and context similarity statistics. The majority of DSMs adopt high-dimensional representations, while the underlying space geometry is not explicitly taken into consideration during the design of algorithms aimed for performing several semantic tasks.

Following the *low-dimensional manifold* hypothesis we propose to extend distributional semantic models (DSMs) into a hierarchical model of *manifolds* (or concepts) that contain related words. Global operations on the lexical space are decomposed into local operations on the low-dimensional sub-manifolds [18]. Our goal is to exploit this hierarchical low-rank model to estimate relations between words, such as semantic similarity. The proposed low-dimensional manifold DSM is constructed in four steps: 1) fragmentation of the lexical space by identifying domains that correspond to the low-dimensional manifolds, i.e., groups of words that are related with some kind of relation, for example groups with semantically similar words, 2) creation of a sparse connectivity graph among words of each manifold 3)

[1] A corpus is a large text of sentences.

construction of a DSM for each manifold by representing each manifold as a low dimensional space, (i.e., run the dimensionality reduction algorithm to each manifold) 4) combine the manifold DSMs to come up with global measures of lexical relations.

A variety of algorithms can be found in the literature for projecting a set of tokens into low dimensional sub-spaces, given a token similarity or dissimilarity matrix. Depending on the nature of the dataset, these projection algorithms may or may not preserve the local geometry of the original dataset. Most dimensionality reduction algorithms make the implicit assumption that the underlying space is metric, e.g., Multidimensional Scaling (MDS) [19] or Principal Component Analysis (PCA) [20] or the ones using non-negative matrix factorization [21] and typically fail to capture the geometry of manifolds embedded in high dimensional spaces. A variety of dimensionality reduction algorithms have been developed that respect the local geometry. Some examples are the Isomap algorithm [22] that performs the projection based on a weighted neighborhood graph, Local Linear Embedings (LLE) [23] that assigns neighbors to each data point, Random Projections [24], [25] that preserves the manifold geometry by executing random linear projections and others (Hessian Eigenmaps (HLLE) [26]; Maximum Variance Unfolding (MVU) [27]). The *manifold hypothesis* has also been studied by the representation learning community where the local geometry is disentangled from the global geometry mainly by using neighborhood graphs [28] or coding schemes [29]. For a review see [30].

A fundamental problem with all aforementioned methods when applied to lexical semantic spaces is that they do not account for ambiguous tokens, i.e., word senses. The main assumption of dimensionality reduction and manifold unfolding algorithms is that each token (word) belongs to a single sub-manifold. This in fact is not true for polysemous words, for example the word 'green' could belong both to the domain *colors*, as well as to the domain *plants*. In essence, lexical semantic spaces are manifolds that have singularities: the manifold collapses in the neighborhood of polysemous words that can be thought of *semantic wormholes* that can instantaneously transfer you from one domain to another. Our proposed solution to this problem is to *allow words to live in multiple sub-manifolds*.

The algorithms proposed in this work are build on recent research work on distributional semantic models and manifold representational learning. Manifold DSMs can be trained directly from a corpus and do not require a-priori knowledge or any human-annotated resources (just like DSMs). We show that the proposed low-dimensional, sparse and hierarchical manifold representation significantly improves on the state-of-the-art for the problem of semantic similarity estimation.

The following paper have been written in relation to this thesis work [18].

1.3 Thesis Outline

The thesis is organized as follows: Chapter 2 gives a brief review of different approaches of DSMs and of the applications that they apply. Chapter 3 presents the motivation, the system architecture and the analysis of the different modules of the DSM system proposed in this work. Chapter 4 displays the experimental procedure and the results. Thesis is concluded at Chapter 5.

Chapter 2

Distributional Semantic Models (DSMs)

Semantic similarity metrics can be broadly divided into the following types: (i) metrics that rely on knowledge resources (e.g., WordNet), and (ii) corpus-based that do not require any external knowledge source. A detailed review of the major WordNet-based metrics can be found in [31]. Corpus-based metrics are formalized as Distributional Semantic Models (DSMs) [15] based on the distributional hypothesis of meaning [16]. A detailed review about semantic similarity metrics and DSMs can be found in [32]. In this chapter a brief analysis of corpus-based metrics is presented, that is mainly focused on the methodologies that are relevant to our work.

2.1 Word Embeddings

Word embeddings constitute a research area of deep learning community, that was introduced in [33]. Word embeddings refer to the representation of vocabulary words, $\mathcal{W}$, of some language, with vectors. Actually, a parameterized function maps the words of vocabulary to high-dimensional vectors (usually 200 to 500 dimensions). So, for example words "university" and "department" may be represented as:

- "university" $\rightarrow$ [0.1, -0.0, 0.3, $\cdots$]
- "department" $\rightarrow$ [0.2, 0.1, -0.8, $\cdots$]

Generally, word embeddings are expected to capture the attributional similarities [34] between vocabulary items, meaning that words that appear in similar contexts should be close to each other in the projected space, i.e., words with similar meanings should have similar vectors.

Methods to generate this mapping include neural networks [35, 36], representations based on the context in which words appear in text [37] and representations based on

words co-occurrence statistics from text, [38, 39], For instance, in [33] the embedded word vectors are trained over large collections of text using variants of neural networks. This model inspired the creation of other neural language models that eliminate the linear dependency on vocabulary size [40], a hierarchical linear neural model proposed in [41], a recurrent neural network for language modeling architecture was investigated in [42]. Such architectures are trained over large corpora of unlabeled text with the aim to predict correct representations. Linguists suggested that words occurring in similar contexts tend to have similar meanings [43]. Thus, the utilization of words co-occurrence statistics is a fair choice to embed similar words into a common vector space [44]. In such approaches generally word frequencies are calculated from text, afterwards some transformations may be applied, such as PPMI, then the dimensionality of word vectors is reduced usually to 200 up to 500 dimensions and finally similarities scores are extracted. Word embeddings resulting from neural language models have been shown to to improve the performance to various NLP tasks, such as syntactic parsing [45] and sentiment analysis [46]

Recently, in [47] it was shown that that the embeddings created by a Recursive Neural Network (RNN) are able to encode both attributional similarities between words and also similarities between pairs of words. More specifically, they observed regularities between pairs of words sharing a particular relationship, for example, let the vector of word w_i be denoted as $\mathbf{x}_{w_i}$, then regarding the singular-plural relation, they observed that $\mathbf{x}_{apple} - \mathbf{x}_{apples} \approx \mathbf{x}_{apple} - \mathbf{x}_{apples}$ or $\mathbf{x}_{family} - \mathbf{x}_{families} \approx \mathbf{x}_{apple} - \mathbf{x}_{apples}$ and that this idea is extended to a variety of semantic relations, such as the male-female relation, i.e., $\mathbf{x}_{woman} - \mathbf{x}_{man} \approx \mathbf{x}_{queen} - \mathbf{x}_{king}$. This, denotes that maybe there exist different dimensions in lexical space for such semantic relations, so probably there exist a gender dimension or another for singular vs plural. Thus, it is seems that neural networks handle to automatically and efficiently represent the data and the relationships among the data objects.

2.2 Metrics of Semantic Similarity Using Web Documents

DSMs can be distinguished into (i) unstructured: use bag-of-words model [48] and (ii) structured: exploitation of syntactic relationships between words [15, 49]. The Vector Space Model (VSM) constitutes the main implementation for both unstructured and structured DSMs. More specifically, a representational vector is built for each word and then various of metrics can be applied for the computation of similarity between those vectors, for example

cosine similarity constitutes a measurement of word similarity that is widely used on the top of VSM, where the similarity between two words is estimated as the cosine of their respective vectors whose elements correspond to corpus-based statistics. Here, we will present the methodology of building such unstructured DSMs.

2.2.1 Corpora Creation

For the computation of text-based semantic similarity between words, a corpus (i.e, a large text) is utilized. There are many corpora that can be directly downloaded from web, such as the SemCor3 corpus [1], but also one could built new corpora. The creation of new corpora gives several advantages to the users, most important is the fact that more domain specific corpora can be built, for example one corpus may contain mainly sentences concerning medical issues, also it can be built in such a way that it will contain only information that is relevant to a given vocabulary. Bellow we describe one well known methodology that may be used for the creation of corpora from web [2]. As first step, the definition of a vocabulary in a specific language is required. Then, for each word of vocabulary an individual query is formulated that contains only the word itself, where the term query refers to the 'search text' that is sent to a web searching engine, such as, Google [3] or Yahoo! [4]. Afterwards, each query is sent separately to Yahoo! Search API [5] (or another available API for developers), and then only the top ranked documents/results, e.g., 1000 top documents, are extracted from each query. Thereafter, from each document, its snippet is extracted, where the term snippet refers to the small paragraph (usually two lines) under the URL of the result that usually describes each document. Finally, all snippets from all queries are merged, resulting a corpus relevant to our vocabulary. In Fig. 2.1 an example of results from Yahoo! search engine is presented as well as the visualization of terminology. There are many other techniques that someone could use, for example one could add semantically similar words to a web query (query expansion) in order to increase the relevance of retrieved documents or experiment with AND, OR queries or fuse differently the web results [50–52].

[1] http://www.cse.unt.edu/rada/downloads.html
[2] This methodology is also used from us in order to create the corpora.
[3] http://www.google.gr
[4] http://www.yahoo.com
[5] http://developer.yahoo.com/search/

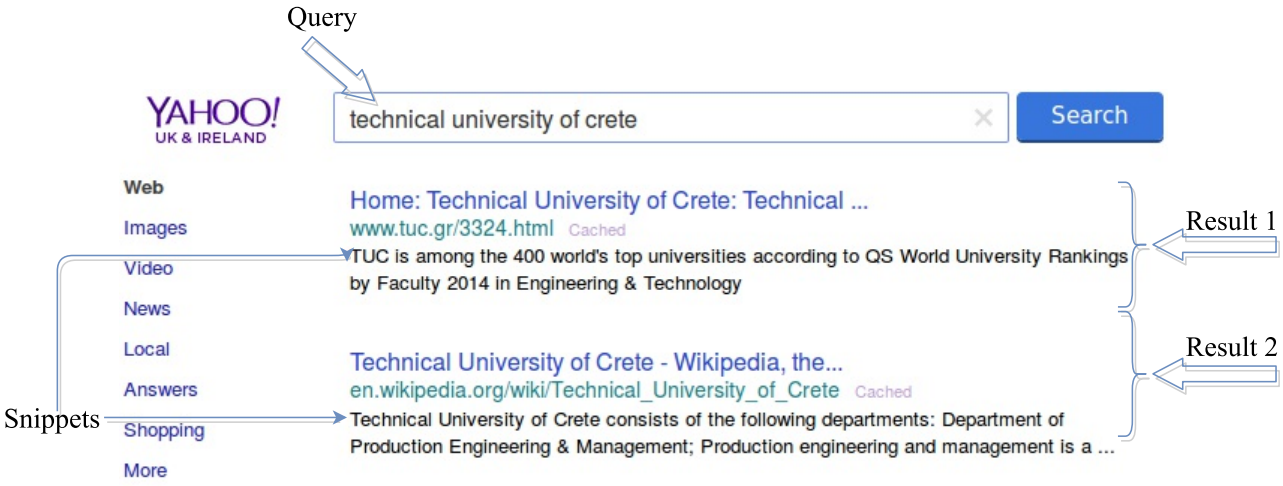

Figure 2.1: Example of results from Yahoo! search engine and visualization of terminology.

Let the following sentences serve as a toy example of a corpus whose lexical content is assumed to capture the semantics for the target words, where each sentence correspond to a different snippet.

```
· TUC is among the 400 world's top universities according to QS World
University Rankings by Faculty 2014 in Engineering & Technology
· Technical University of Crete consists of the following departments:
Department of Production Engineering & Management
· Welcome to the website of the School of Environmental Engineering,
Technical University of Crete
· University of Crete is one of the top universities in the world,
The Times Higher Education, World University Rankings grades the top
universities in the world
```

2.2.2 Context-based Statistics

Following the distributional hypothesis of meaning [16], which implies that semantic similarity between words is a function of the overlap of their linguistic contexts, sentences or even few words in the left and right context of a target word of vocabulary may be utilized in order to built its semantic representation [53]. More specifically, the context words of "crete" in the sentence "university of crete is one of the top universities", with contextual

window 2, are the "university", "of", "'is" and "one". So, when the target vocabulary, $\mathcal{W}$, is defined, as well as, the 'context vocabulary', $\mathcal{W}_c$, that may contain all the corpus words (or corpus sentences or the top frequent corpus words or other useful contextual features), then context-based statistics may be computed as follows. For each target word $w_i \in \mathcal{W}$, with $i = 1, \cdots, |\mathcal{W}|$, a representational vector, $\mathbf{c}^i$, is built, with $|\mathbf{c}^i| = |\mathcal{W}_c|$. This representational vector may contain various statistical measurements, that concerns the word w_i and the contextual features $v_k \in \mathcal{W}_c$, with $k = 1, \cdots, |\mathcal{W}_c|$, for example the times that each v_k occurs as context of w_i in the corpus or an indicator function if v_k occurs as context word of w_i or other more complex statistics [32]. An example of context-based representational vectors built for vocaculary words "technical", "engineering" using the toy corpus of Sec. 2.2.1 , with contextual window 3 and the indicator function as statistical measurement, is presented in Table 2.1.

$\mathcal{W}$ \ $\mathcal{W}_c$	university	crete	school	$\cdots$	environmental
technical	1	1	0	$\cdots$	1
engineering	1	0	1	$\cdots$	1

Table 2.1: Example of context-based representational vectors.

Dimensionality Reduction

The DSMs approaches proposed in literature mainly built one high dimensional representation of lexical space; either by utilizing the representational vectors of vocabulary words as is or by projecting those vectors to a space of lower dimensions (usually 50 up to to 2000 dimensions may be used) [15, 54–56]. The most widely-used techniques for reducing the dimensions of such matrices/vectors is the Principal Component Analysis (PCA) [4, 57, 58] and the MultiDimensional Scaling (MDS) [19, 59, 60]. Thus, a representational vector of one word w_i, $\mathbf{c}^i$, that is $|\mathcal{W}_c|$-dimensional may be is transformed to another representational vector of much lower dimensions. The low-dimensional approximation is considered to (i) capture the latent meaning of words, (ii) reveal higher-order co-occurrences, (iii) reduce to the "noise" introduced by non-informative contextual features, and (iv) tackle the sparsity problem. A detailed analysis regarding the semantic representations and the dimensionality of semantic VSMs is presented in [44].

Metrics of Semantic Similarity

Context-based semantic similarity metrics rely on the distributional hypothesis of meaning according to which the semantic similarity is implied by the paradigmatic relatedness. Cosine similarity is reported to be the most widely-used similarity metric with respect to VSM [44, 53]. Let $\tilde{\mathbf{c}}^i, \tilde{\mathbf{c}}^j$ correspond to the representational vectors of two words w_i and w_j, then the similarity is estimated as the cosine of their representational vectors, as follows [48]:

$$S_c(w_i, w_j) = \frac{\sum_{k=1}^{m} \tilde{\mathbf{c}}^i(k) \cdot \tilde{\mathbf{c}}^j(k)}{\sum_{k=1}^{m} \tilde{\mathbf{c}}^i(k)^2 \cdot \sum_{k=1}^{m} \tilde{\mathbf{c}}^j(k)^2} \tag{2.1}$$

where m is the dimensionality of representational vectors.

A variety of other metrics or alternatives [61] have been applied for estimating the semantic similarity between words based on their representational vectors, a more detailed analysis can be found here [32]. Another approach is to transform the representational vectors to probability distributions, using n-gram language modeling, and then apply metrics such as the Kullback–Leibler distance that measures the dissimilarity between the two distributions, this idea was employed to [62, 63].

2.2.3 Co-occurrence-based Statistics

Co-occurrence-based statistics can also be extracted for the estimation of semantic similarity, the underlying assumption of co-occurrence-based metrics is that two words that co-exist in a specified context are semantically related. Thus, this approach utilize the association ratios between words that are computed using their co-occurrence frequency in a specified context. The exploitation of direct (i.e., first-order) co-occurrence statistics constitutes the simplest form of unstructured DSMs. A key parameter for such models is the definition of the context in which the words of interest co-occur: from entire documents [64] to paragraphs [65] and sentences [17]. The effect of co-occurrence for the task of similarity computation between nouns is discussed in [17].

Let $\{M\}$ be a set of such contexts. e.g., sentences, while $\{M; w_i, \ldots, w_{i+n}\}$ stands for the number of occurrences of words $w_i, \ldots, w_{i+n}$ within $\{M\}$. A widely-used measurement, proposed in [66, 67] and motivated by Kolmogorov complexity, is the 'normalized Google distance' and is defined as follows:

$$G_0(w_i, w_j) = \frac{A - \log |M; w_i, w_j|}{\log |M| - B} \tag{2.2}$$

where $A = \max\{\log |M; w_i|, \log |M; w_j|\}$ and $B = \min\{\log |M; w_i|, \log |M; w_j|\}$

This metric is a dissimilarity measure, i.e., as the semantic similarity between two words increases the metric takes smaller values. The scores assigned by (2.2) are unbounded, ranging from 0 to ∞. In [68], a variation of the normalized Google distance was used, proposing a bounded similarity measure called 'Google-based semantic relatedness', defined as:

$$G(w_i, w_j) = e^{-2 \cdot G_0(w_i, w_j)} \tag{2.3}$$

where $G_0(w_i, w_j)$ is computed according to (2.2). The Google-based semantic relatedness is bounded in [0,1] [6].

Other metrics that can be employed for the computation of co-occurrence similarity are the 'Jaccard' coefficient that can be applied to the specific problem as:

$$J(w_i, w_j) = \frac{|M; w_i, w_j|}{|M; w_i| + |M; w_j| - |M; w_i, w_j|} \tag{2.4}$$

where if w_i and w_j are the same word then the Jaccard coefficient is equal to 1 (absolute semantic similarity) and if the two words never co-occur then Jaccard coefficient equals to 0. Also, the 'Dice' coefficient that is related to the Jaccard coefficient and is computed as:

$$C(w_i, w_j) = \frac{2 \cdot |M; w_i, w_j|}{|M; w_i| + |M; w_j|} \tag{2.5}$$

Again, one could experiment with random variables, assuming that the number of documents (or snippets or paragraphs, etc) indexed by w_i, w_j are random variables and utilize the the pointwise Mutual Information (MI) to measure the mutual dependence between the occurrence of words w_i, w_j [69].

$$I(w_i, w_j) = log \frac{\frac{|M; w_i, w_j|}{|M|}}{\frac{|M; w_i|}{|M|} \cdot \frac{|M; w_j|}{|M|}} \tag{2.6}$$

This measurement quantifies how the knowledge of one variable reduces the uncertainty about the other.

[6] In this work, this metric was adopted based on its good performance in word-level semantic similarity tasks [17], where the co-occurrence of words was defined at snippet-level.

2.2.4 Network-based Metrics

Very few approaches have been proposed in literature that focus to the creation of more sophisticated representations of lexical space. Recently, motivated by the graph theory, several aspects of human languages have been modeled using network-based methods. In [70, 71] an overview of network-based approaches is presented for a number of NLP problems. Network-based representations constitute a promising idea that outperformed for the task of semantic similarity estimation [17].

A brief summary of the network-based DSM proposed in [17] is the following. Firstly, a web harvested corpus of snippets is built. Afterwards a semantic network is constructed encoding the semantic relations between words in corpus, where co-occurrence and context features are used to measure the strength of relations. The network is a parsimonious representation of the information encoded in corpus. Thus, each word that is included in lexicon may be represented by a subgraph that is referred as the semantic neighborhood of word. An example of semantic neighborhoods of words "fruit", "forest" and "plant" is depicted in Fig. 2.2. Thereafter, three similarity metrics are proposed that are built on the top of those graphs and exploit the neighborhoods, namely the 'maximum similarity of neighborhoods', the 'correlation of neighborhood similarities' and the 'sum of squared neighborhood similarities'.

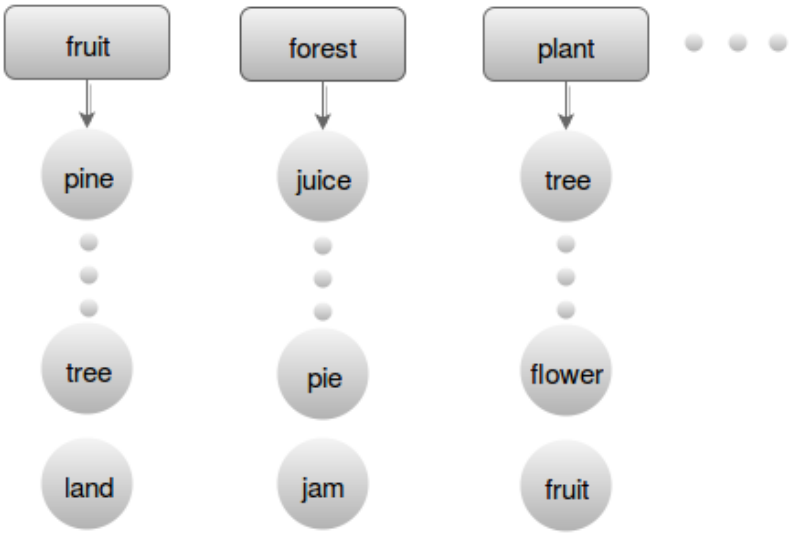

Figure 2.2: Example of semantic neighborhoods representing words "fruit", "forest" and "plant".

The DSM proposed in [17] inspired us to focus in the area of representation of lexical space and not in the area of technical methodologies regarding the estimation of semantic similarity of words (for example methodologies such as the corpus creation, alternative methods for counting word frequencies, the exploitation of contexts words or of syntactic information and other technical aspects). As a result, an alternative top-down hierarchical manifold representation is proposed in Chapter 3 that respects the cognitive principles and is an ensemble of parallel, sparse and low-dimensional subspaces.

2.3 Applications

The estimation of semantic similarity between words, sentences and documents is the building block of many research disciplines, such as computational linguistics [8], semantic web [9] and artificial intelligence [10, 11]. There exist a variety of applications for semantic similarity, both at word and sentence level. For example, problems highly related to semantic similarity is the paraphrasing which is bidirectional and based on semantic equivalence [72], and the textual entailment which is directional and based on relations between semantics [73, 74]. Other examples, include machine translation [75], information extraction [76] and question answering [77].

The recent years affective text analysis is another very hot research area. Based on hypothesis that "semantic similarity can be translated to affective estimates", there have been proposed affective systems [8, 78] that model the mapping between semantic and affective space and achieve very high performance. More specifically, given a set of vocabulary words, semantic similarity ratings have been extraxted between vocabulary words as first step, afterwards given the existance of ground truth affective ratings for a small subset of vocabulary words (known as "seeds"), a model is trained to compute the affective score for each word of vocabulary as the algebraic combination of the semantic similarities and the affective ratings of seed words. The same idea can also be applied for the estimation of other dimensions of words (besides affective dimension). As described in [79], semantic similarity can be mapped to other word representations such as, the concreteness level of a word (i.e., that measures how concrete is one word, for example the term 'table' is very concrete while the term 'liberty' is totally abstract), the imagability level of a word (i.e., that qualifies how much the "hearing" of a word triggers us to imagine images), the familiarity level (i.e., how familiar a word is) and the age of acquisition level (i.e., that qualifies the age that a human is able to acquire one word, for example).

Spoken dialogue systems is another application where the influence of semantic similarity is significant. Grammar induction is a fundamental part of spoken dialogue systems and is highly dependent on the availability of semantic classes that correspond to domain concepts, where examples of domains include the travel domain, the medical domain, the finance domain and other, and one domain concept correspond to a class that include words or phrases of similar meaning. The creation of such classes is based on semantic similarity between the candidate terminals that is usually followed by a clustering algorithm for the construction of the classes [80].

Chapter 3

LDMS: Manifold-based Distributional Semantic Model

In this chapter we will describe the Manifold-based Distributional Semantic Model proposed in this thesis. Firstly, we will present some ideas that motivated us to built this system and secondly we will present and analyze whole system and its components.

3.1 Motivation

Consider a finite metric space $(\mathcal{M}, d)$ and the the power set of $\mathcal{M}$, $\mathcal{P}(\mathcal{M})$, which is also finite. For any $\mathcal{A}, \mathcal{B} \in \mathcal{P}(\mathcal{M})$ the common sense set distance is defined as:

$$d_s(\mathcal{A}, \mathcal{B}) = \min_{a_i \in \mathcal{A},\ b_j \in \mathcal{B}} d(a_i, b_j). \tag{3.1}$$

It easy to see that we may find triplets $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathcal{P}(\mathcal{M})$ where the triangle inequality does not hold. Thus, under the "common sense" set distance the power set space $\mathcal{P}(\mathcal{M})$ is not metric, i.e., $(\mathcal{P}(\mathcal{M}), d_s)$ is not a metric space, hence, the nice convergence properties of metric spaces and the resulting notion of neighborhood are not satisfied. Although these notions might not exist globally in the power set space, they are satisfied locally under some assumptions. For instance, if $\mathcal{M}$ has n elements, $\xi_1, \xi_2, \cdots, \xi_n$, then the set space $\mathcal{Q} \triangleq \{\{\xi_1\}, \{\xi_2\}, \cdots, \{\xi_n\}\} \subset \mathcal{P}(\mathcal{M})$ is a metric space under d_s.

To demonstrate that the power set space, $(\mathcal{P}(\mathcal{M}), d_s)$, is not metric in general, consider the set $\mathcal{Y} \triangleq \{\{\xi_1, \xi_2\}, \{\xi_3\}, \cdots, \{\xi_n\}\} \subset \mathcal{P}(\mathcal{M})$, i.e., $\mathcal{Y}$ consists of a single set with two elements $\mathcal{A} = \{\xi_1, \xi_2\}$ and the single element sets of all the remaining members of $\mathcal{M}$, i.e., $\{\xi_3\}, \{\xi_4\}, \cdots, \{\xi_n\}$. Then, the triangle inequality is satisfied for all triplets of elements of $\mathcal{Y}$, with the possible exception of triplets containing set $\mathcal{A}$, where it may not holds.

For instance, an illustrative example is depicted in Fig. 3.1 with $n = 4$, $\mathcal{A} = \{\xi_1, \xi_2\}$,

$\mathcal{B} = \xi_3$, and $\mathcal{C} = \xi_4$, where we have:

$$d_s(\mathcal{B}, \mathcal{C}) > d_s(\mathcal{A}, \mathcal{B}) + d_s(\mathcal{A}, \mathcal{C}) \tag{3.2}$$

The probability of violating the triangle inequality is higher when the two members of $\mathcal{A}$ are far away in the underlying metric space $(\mathcal{M}, d)$, i.e., $d(\xi_1, \xi_2) > 2\delta$ and in addition, ξ_3 and ξ_4 are in the δ-neighborhood of ξ_1 and ξ_2 , respectively (or vice versa), i.e., $d(\xi_1, \xi_3) < \delta$ and $d(\xi_2, \xi_4) < \delta$.

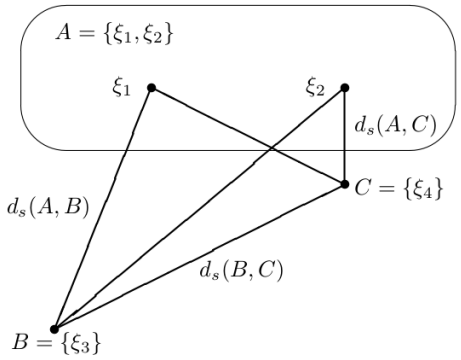

Figure 3.1: Example of non metric power set space, $(\mathcal{P}(\mathcal{M}), d_s)$.

In the context of word semantics, we assume that the conceptual space, $\mathcal{M}$, defined as the set containing all atomic elements word senses, i.e., $\mathcal{M} = \{\xi_1, \xi_2, \xi_3, \xi_4, \cdots, \xi_n\}$, together with the metric of semantic similarity $S(.)$ (normalized in [0,1]), forms a metric space with distance $1 - S$, i.e., $(\mathcal{M}, 1 - S)$.

Then, words may be considered as sets of word senses. Namely, the set of words, $\mathcal{W}$, is a subset of the power set of $\mathcal{M}$, i.e., $\mathcal{W} \subset \mathcal{P}(\mathcal{M})$, with word level semantic similarity defined in space $\mathcal{W}$ as follows (maximum sense semantic similarity assumption):

$$S_{\mathcal{W}}(w_A, w_B) = \max_{\xi_i \in w_A, \eta_j \in w_B} S(\xi_i, \eta_j) \tag{3.3}$$

where w_A, w_B two arbitrary words.

More specifically, in Fig. 3.1 let the word w_A, represented by the set $\mathcal{A}$, consisting of just two word senses (or concepts), ξ_1 and ξ_2, while words w_B and w_C, two monosemous words with a single word sense each, ξ_3 and ξ_4, respectively. For example, let the polysemous word $w_A =$*'book'* with corresponding word senses $\xi_1 =$*'book with the sense of reservation'* and $\xi_2 =$*'book with the sense of reading'* and the monosemous words $w_B =$'*travel*' and

w_C='*author*'. Then, apparently the triangle inequality is violated for the pair ('travel', 'author').

The above example reveals that the space $\mathcal{W} = (\mathcal{W}, 1 - S_{\mathcal{W}})$ does not form a metric space because we found an example that does not satisfy the triangle inequality. However, due to the fact that $d(\xi_1, \xi_2) = 1 - S(\xi_1, \xi_2) > 0$, there exists an ϵ-neighborhood around ξ_1, that contains only word sense ξ_1. Namely, there exists ϵ, such that $\xi_2 \notin \mathcal{B}(\xi_1, \epsilon) = \{\xi \in \mathcal{M} : 1 - S(\xi_1, \xi) < \epsilon\}$. The same reasoning can be applied to all other words $w \in \mathcal{W}$, i.e., for any word a neighborhood can be found around each sense of the word such that it does not contain the other senses of the word. Clearly, for any word $w \in \mathcal{W}$ and any $\xi_i \in w$, all such balls $\mathcal{B}(\xi_i, \epsilon_i)$ are subsets of $\mathcal{M}$ and obviously, subsets of the power set of $\mathcal{M}$. However, $\mathcal{B}(\xi_i, \epsilon_i)$ may not be subset of $\mathcal{W}$, since $w \backslash \mathcal{B}(\xi_i, \epsilon_i) \notin \mathcal{W}$.

It is worth emphasizing that in the case where $\mathcal{B}(\xi_i, \epsilon_i)$, contains ξ_i and at most one word sense from some of the other words of $\mathcal{W}$, then $(\mathcal{B}(\xi_i, \epsilon_i), 1 - S_{\mathcal{W}})$ forms a metric space. Thus, in such scenario, $\mathcal{W}$ is locally metric but globally non-metric. In essence, what the above analysis suggests is that for each monosemous word we can hope for metricity in a neighborhood around this word, while for a polysemous word we can hope for metricity in a neighborhood around each of its senses.

Regarding the dimensionality of lexical space, the global lexical semantic space is expected to be high-dimensional, but organized in such a way that the significant semantic relations can be exported from manifolds of much lower dimensionality embedded in this high dimensional space [12]. We assume that (at least some of) these sub-manifolds contain semantically similar words (or word senses). For example, a potential sub-manifold in the lexical space could be the one that contains the colors (e.g., *red, blue, green*). But in fact many words, such as *book, green, fruit*, are expected to belong simultaneously in semantically different manifolds because they have multiple meanings. A simple example of a method to bootstrap the manifold recreation process is to build a manifold around each word, i.e., *the semantic neighborhood of each word defines a manifold*. For example, in Figure 3.2 we show the semantic neighborhood of *fruit*, where the neighborhood is built just by selecting the top most similar neighbors of word *fruit*.

The connections between words indicate high semantic similarity, i.e., this is a pruned semantic similarity graph of all words in the semantic neighborhood of the word 'fruit'.

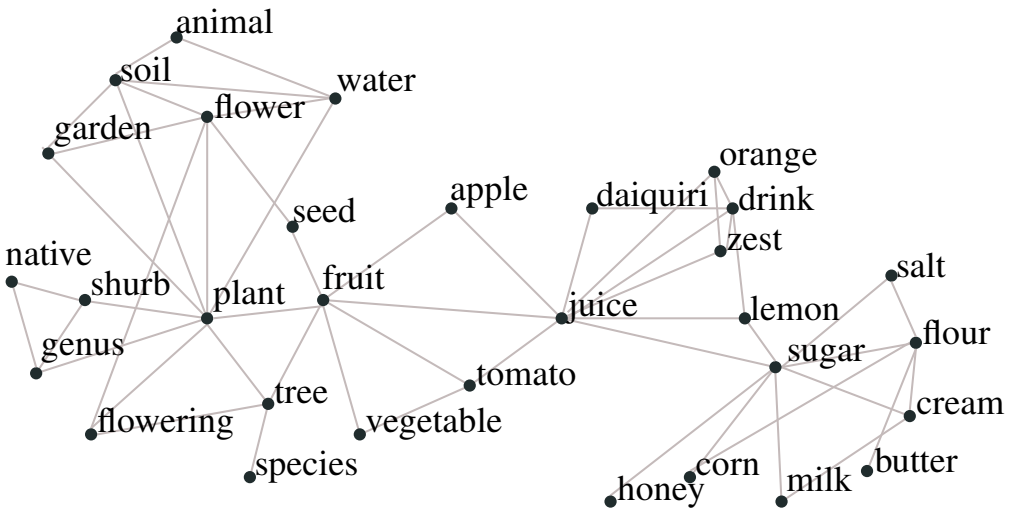

Figure 3.2: Visualization of the semantic neighborhood of the word 'fruit'.

It is clear from this example that in a typical neighborhood there exist word pairs that should be 'connected' to each other because they have close semantic relation, like {*flower*, *plant*} and others that should not be 'connected' because they are semantically apart, like {*garden*, *salt*}. A *sparse encoding* of the semantic similarity relations in a neighborhood is needed in order to achieve (via multi-dimensional scaling) a parsimonious representation with good geometric properties[1].

3.2 System Architecture

The proposed end-to-end Low-dimensional manifold DSM (LDMS) system is depicted in Figure 3.3. and is composed from the following parts:

1. **Construction of manifolds**: This step constitutes the identification of manifolds, $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_{|\mathcal{V}|}\}$. Basically each manifold is a set of items/words connected to each other with some kind of relation, for example one could utilize semantic similarity as relational function and construct different manifolds with semantically similar words. Note that one word may belong to many manifolds, thus the ambiguity property of the lexical space may be implicitly imported to the model. We propose

[1]Compare for example with Isomap [22] were a short- and long-distance metric is used. When using sparse encoding the long-distance metric is set to a very large fixed number (similarity set to 0). In both cases, the underlying manifold is unfolded and low-dimensional representation with (close to) metric properties are discovered.

two different methods for constructing the manifolds, that are described in Sec. 3.2.1, but many other ideas could also be implemented.

2. **Sparse encoding of manifolds**: In lexical space the connections between words are assumed to be very sparse, even if we are dealing with very small sets, such as the neighborhood of word fruit in Fig. 3.2. In order to model this property of lexical space, this step deals with the automatic construction of a sparse connectivity graph among items/words in each manifold, resulting a set of sparse connectivity matrices each one corresponding to one manifold $\tilde{\mathcal{S}} = \{\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2, ..., \tilde{\mathbf{S}}_{|\mathcal{V}|}\}$. The graph connectivity or sparseness matrix identifies the word pairs that should be encoded in a manifold and is defined as $\tilde{\mathbf{S}}_k \in \mathbb{B}^{n \times n}$, with $k = 1, ..., |\mathcal{V}|$, where value 1 indicates that the word pair is encoded and 0 indicates that the pair is ignored. Note that one could ignore the sparsity modeling just by assigning 1 to all connections. The methods that we propose for the construction of sparse connectivity matrices are described in Sec. 3.2.2.

3. **Low dimensional representation of manifolds**: The lexical space is assumed to consists from manifolds of *very low dimensionality*. Thus, in this step all the manifolds are projected in a low-dimensional space. In Sec. 3.2.3 we propose a dimensionality reduction algorithm that encounters the sparse connectivity matrices in order to perform the projection of the manifolds and constitutes an alternation of MDS.

4. **Fusion from different subspaces**: After this hierarchical low-rank model is built, one could exploit those representations to estimate relations between words. In this work, we estimated the semantic similarity as described in Sec. 3.2.4.

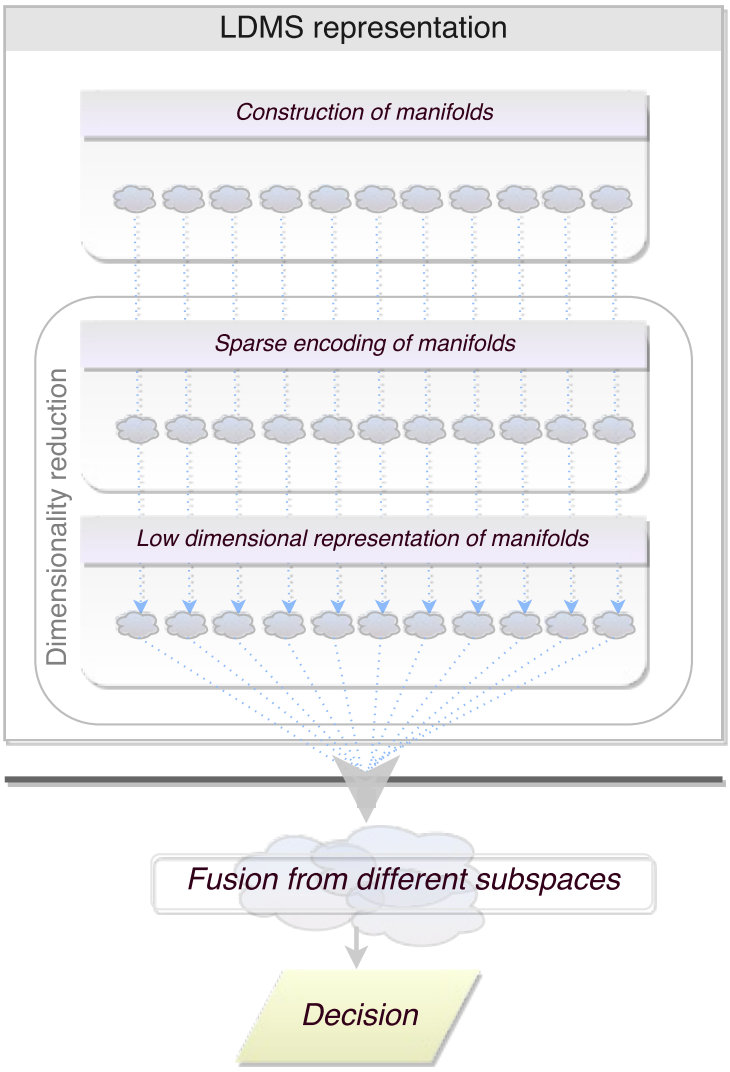

Figure 3.3: System architecture of LDMS.

Note that the proposed representation could be applied and to other spaces with complex properties, like the lexical space. More specifically, the aspects that are implicitly or explicitly modeled in this system are: a manifold based space, sparse, global non metric but local metric and ambiguous.

3.2.1 Construction of Manifolds

In this section we will describe two methods for the construction of manifolds, namely the neighborhoods and the hierarchical categories, where the first approach is unsupervised,

i.e., no manually annotated data are required and the second approach requires a manually annotated resource.

Construction of Manifolds: Neighborhoods

This method requires the availability of a global proximity matrix [2], among items, i.e., vocabulary words, $\mathcal{W}$, in our case. The manifolds $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_{|\mathcal{V}|}\}$ are created as follows: for each word of vocabulary a corresponding manifold is created, so in this approach the vocabulary size is equal to the cardinality of manifold set, i.e., $|\mathcal{W}| = |\mathcal{V}|$. More specifically, for each $w_k \in |\mathcal{W}|$, with $k = 1, ..., |\mathcal{V}|$, $\mathcal{V}_k$ will consists of words that are highly similar to w_k, i.e., the k^{th} manifold will be the semantic neighborhood of word w_k. Thus, each manifold $\mathcal{V}_k$ is created by selecting the top n most semantically similar words of w_k based on the (global) similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{W}| \times |\mathcal{W}|}$, where $\mathbf{S}$ can be estimated by using any of the baseline semantic similarity metrics presented in Section 2.2.2. We have experimented with various manifold sizes n ranging between 20 and 200 neighbors; note that each word w_k may belong to multiple domains, for example w_k may belong to $\mathcal{V}_k$ and to other manifolds constructed from other words. Afterwards, for each manifold $\mathcal{V}_k$ a separate DSM will be built. This approach is completely unsupervised and does not require any manually annotated recourse, thus it can be easily ported to many languages.

Construction of Manifolds: Hierarchical Categories

The main idea behind this method is the exploitation of a hierarchal information regarding the vocabulary set, $\mathcal{W}$, in order to built the manifolds. Here the hierarchy provided by WordNet is utilized, where WordNet is a large lexical database of English words, where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. In Fig. 3.4 a toy example of hierarchy structure provided by WordNet is depicted. Actually, it corresponds to a tree-like structure where the top category of everything is the 'entity' and thereafter the tree expands to lower levels into other sub-categories and words. For example, in the figure two sub-categories are the 'physical entity' and 'abstract entity' which are also expanded to lower levels. Note that, ambiguous words will be assigned to different parts (or and different levels) of the tree structure for example the word 'state' is mapped under the 'region' and also under the

[2]Proximity denotes either similarity or distance. Let's assume that a global semantic similarity matrix is provided, by using for example one of the methods described in Chapter. 2. In our experiments, the Google-based Semantic Relatedness was employed using a web-harvested corpus of document snippets.

'attribute'.

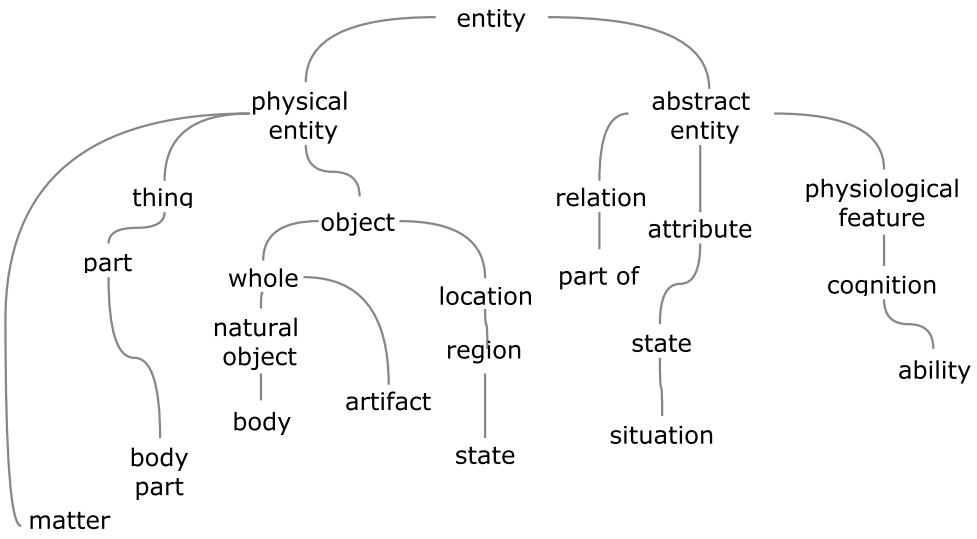

Figure 3.4: Example of the structure of WordNet hierarchy.

The manifolds $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_{|\mathcal{V}|}\}$ are created as follows: for each word of $\mathcal{W}$, we have marked the categories, the super-categories, the super-super-categories, etc., assigned to the word, until we reach the root category 'entity'. For example, let the vocabulary word 'dog', then the categories that are marked from word 'dog' until the root category 'entity', are the 'canine','carnivore','placental','mammal', 'animal',..., 'entity'. Thus, we are able to extract a set of different categories, $\mathcal{C}$, from all vocabulary words as well as the information of which words of $\mathcal{W}$ are assigned to each one of the categories, in the example above the word 'dog' is assigned to all the aforementioned categories. Thereafter we are able to built one manifold for each category, by adding to each manifold the words that are assigned to the corresponding category. Thus, the number of manifolds will equal to the number of the corresponding categories, while the size of the different manifolds will not be equal.

3.2.2 Sparse Encoding of Manifolds

Here, we propose different approaches for the automatic construction of sparse connectivity matrices among words of each manifold, $\tilde{\mathcal{S}} = \{\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2, ..., \tilde{\mathbf{S}}_{|\mathcal{V}|}\}$. Given a global similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{W}|}$, and the manifolds, $\mathcal{V}_k$ with $k = 1, ..., |\mathcal{V}|$, the corresponding sub-similarity matrices, $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{|\mathcal{V}|}\}$, of the manifolds can be easily build since the words included in each manifold are subset of vocabulary $|\mathcal{W}|$. So, let $i, j, z = 1, \cdots, |\mathcal{V}_k|$

correspond to the indexing of words contained in manifold $\mathcal{V}_k$, then the methods for constructing the connectivity graph are the following:

1. **Similarity 0**: Let assume that the global similarity values of $\mathbf{S} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{W}|}$ range in [0, 1], then the word pairs (w_i^k, w_j^k) with corresponding similarity value equal to zero are penalized. More specifically, if similarity of pair (w_i^k, w_j^k) is zero, then the pair will not be 'connected' in the graph of $\mathcal{V}_k$ manifold and 0 will be assigned to $\tilde{\mathbf{S}}_k(i,j)$, else the pair will be 'connected' and 1 will be assigned to $\tilde{\mathbf{S}}_k(i,j)$.

2. **Percentage**: We define the degree of sparseness as the percentage of 0's in matrix $\tilde{\mathbf{S}}_k$. Word pairs (w_i^k, w_j^k) with small similarity values (or equivalently large semantic distance) are penalized and zero values are assigned to their corresponding position (i,j) in $\tilde{\mathbf{S}}_k$ matrix. In essence, the matrix $\tilde{\mathbf{S}}_k(i,j)$ is obtained by hard thresholding on the similarity matrix $\mathbf{S}_k$: all values that are under a threshold are set to 0, while all values equal or greater to the threshold are set to 1. Let $n = |\mathcal{V}_k|$ be the number of words under investigation, then the number of word pairs is $q = \frac{n\cdot(n-1)}{2}$, also let $0 \leq p \leq 1$ be a given percentage score, then the sparseness is defined by penalizing all the $p \cdot q$ less similar word pairs of manifold.

3. **Triangle inequality**: All the word pairs (w_i^k, w_j^k) that do not respect the triangle inequality are identified and 0 is assigned to the corresponding position of connectivity matrix, $\tilde{\mathbf{S}}_k$. More specifically, the penalized pairs are those for which the following statement is true: $\mathbf{S}_k(i,j) < \mathbf{S}_k(i,z) + \mathbf{S}_k(z,j)$ or $\mathbf{S}_k(i,j) = 0$, where w_z^k is another word in manifold $\mathcal{V}_k$.

4. **Triangle inequality target word**: This approach is an extension of the previous one and can be applied only to the case where the the manifolds are build as neighborhoods of words in $\mathcal{W}$, described in Sec. 3.2.1. All the word pairs (w_i^k, w_j^k) that do not respect the triangle inequality regarding only the target word from which the manifold/neighborhood was build are penalized, so if w_z^k correspond to the target word then 0 is assigned to $\tilde{\mathbf{S}}_k(i,j)$ if $\mathbf{S}_k(i,j) < \mathbf{S}_k(i,z) + \mathbf{S}_k(z,j)$ or $\mathbf{S}_k(i,j) = 0$.

5. **Cliques**: The first step of this method includes the construction of the graph's boolean adjacency matrix $\mathbf{A}_k \in \mathbb{B}^{n\times n}$, which is defined based on similarity matrix $\tilde{\mathbf{S}}_k$, where if $\mathbf{S}_k(i,j) > 0$ then $\mathbf{A}_k(i,j) = 1$ and if $\mathbf{S}_k(i,j) = 0$ then $\mathbf{A}_k(i,j) = 0$. Given the graph's boolean adjacency matrix, $\mathbf{A}_k$ we are able to find all maximal cliques on $\mathbf{A}_k$ using the Bron-Kerbosch algorithm in a recursive manner [81, 82]. Thereafter,

the connectivity matrix is defined as follows, if a pair (w_i^k, w_j^k) belong to a maximal clique, then the pair is 'connected' and $\tilde{\mathbf{S}}_k(i,j) = 1$, else $\tilde{\mathbf{S}}_k(i,j) = 0$.

3.2.3 Low Dimensional Representation of Manifolds

In this section, the Sparse Projection (SP) algorithm is described (see also Algorithm 1). SP is the core algorithm for constructing the manifold DSM depicted in Fig. 3.3 and is a dimensionality reduction algorithm that projects a set of n objects/words into a vector space of d dimensions Note that, in the proposed system, SP algorithm is applied to the word manifolds, $\mathcal{V}_k$ with $k = 1, ..., |\mathcal{V}|$, so the notation is formed based on this assumption, but the algorithm could easily be applied to an arbitrary set of objects. The inputs to the algorithm are the dissimilarity or semantic distance matrix $\mathbf{P}_k \in \mathbb{R}^{n \times n}$ where each element $\mathbf{P}_k(i,j)$ encodes the degree of dissimilarity between words w_i and w_j, the connectivity graph of manifold words, $\tilde{\mathbf{S}}_k \in \mathbb{B}^{n \times n}$ and the projection dimension d. The output of SP are the d-dimensional coordinate vectors of the n projected words that form a matrix $\mathbf{X_k} \in \mathbb{R}^{n \times d}$. Each row of matrix $\mathbf{X}_k$, $\mathbf{X}_k(i) \in \mathbb{R}^{1 \times d}$, corresponds to the coordinates of the i^{th} word, w_i^k. Once $\mathbf{X}_k$ is estimated, the dissimilarity matrix can be reconstructed from the low-dimensional space and be transformed to similarity matrix, so the second output of SP is the re-estimated, bounded, sparse, semantic similarity matrix $\hat{\mathbf{P}}_k \in \mathbb{R}^{n \times n}$ of the manifold.

Since the SP algorithm uses as input a dissimilarity or semantic distance matrix, the pairwise manifold word similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{W}| \times |\mathcal{W}|}$ is transformed to a semantic distance (or dissimilarity) matrix $\mathbf{P}_k \in \mathbb{R}^{n \times n}$ as:

$$\mathbf{P}_k(i,j) = c_1 \cdot e^{-c_2 \cdot \mathbf{S}_(i,j)} \tag{3.4}$$

where $c_1, c_2 \in \mathbb{R}$ are constants and the i, j indexes run from 1 to n. In this work, we experimentally chosen $c_1 = c_2 = 20$. The transformation defined by (3.4) was selected in order to non-linearly scale and increase the relative distance of dissimilar words compared to similar ones[3]. As discussed in Section 3.2.2, given a set of words only a small subset of lexical relations should be explicitly encoded between pairs of these words. Therefore, the SP algorithm should only take into account strongly related word pairs and ignore the rest. This is the main difference between our approach compared to the generic MDS algorithm proposed in [19] [4]. Each paragraph that follows corresponds to a module of Algorithm 1.

[3]Similar nonlinear scaling function from similarity to distance can be found in the literature, e.g., [60]

[4]The SP algorithm with 0% degree of sparseness in input connectivity is equivalent to the MDS algorithm.

Random Walk SP: In function MoveWordToDirection(·) of Algorithm 1, the pseudo-variable *direction* z refers to a direction from a standard set of perturbations of each word in the d-dimensional space. For example, if the dimension of the projection is $d = 2$ then the coordinates of each word are modeled as (k_1, k_2), where $k_1, k_2 \in \mathbb{R}$. A potential set of perturbations are the following: $(k_1, k_2 + s)$, $(k_1, k_2 - s)$, $(k_1 + s, k_2)$ and $(k_1 - s, k_2)$, where s is the perturbation step parameter of the algorithm, an example is also depicted in the following figure. For coordinates systems normalized in $[0, 1]^d$ we chose a value of s equal to 0.1. Good convergence properties to global maxima have been experimentally shown for this algorithm setup for multiple runs on (noisy) randomly generated data.

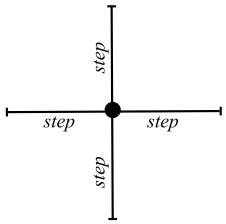

Figure 3.5: Perturbations of an object in $\mathbb{R}^2$.

Error Criterion: The algorithm employs a local and a global error criterion defined as follows:

1. The local error corresponds to the projection error of each individual word w_i^k, where $i = 1...n$ and is defined as the sum of the dissimilarity matrix errors before and after projection computed only for the words that are 'connected' to w_i. Let $\mathbf{D}_k \in \mathbb{R}^{n \times n}$ correspond to the dissimilarity matrix between coordinate vectors, defined as the Euclidean norm [5]:

$$\mathbf{D}_k(i, j) = \|\mathbf{X}_k(i) - \mathbf{X}_k(j)\|_2 \tag{3.5}$$

where $\mathbf{X}_k(i)$, $\mathbf{X}_k(j)$ are the vectors corresponding to words w_i^k, w_j^k, respectively, $i, j = 1, .., n$. The local projection error of word w_i^k is defined as:

$$\text{ComputeLocalError}(\tilde{\mathbf{S}}_k,\mathbf{P}_k,\mathbf{X}_k,i) = \sum_{j=1}^{n} \tilde{\mathbf{S}}_k(i, j) \cdot (\mathbf{D}_k(i, j) - \mathbf{P}_k(i, j))^2 \tag{3.6}$$

[5]Other metrics, e.g., cosine similarity, have also been tested out but Euclidean distance performed somewhat better.

2. The global error of the projection is simply the sum over local errors for all words:

$$e_{tot} = \sum_{i=1}^{n} \text{ComputeLocalError}(\tilde{\mathbf{S}}_k, \mathbf{P}_k, \mathbf{X}_k, i) \tag{3.7}$$

Algorithm 1 Sparse projection (SP)

Require: $\mathcal{V}_k$ // manifold set of n words
Require: $\mathbf{P_k}$ // $n \times n$ dissimilarity matrix
Require: $\tilde{\mathbf{S}}_k$ // $n \times n$ connectivity graph
Require: d // projection dimension
for each word $w_i^k \in \mathcal{V}_k$ **do**
  $\mathbf{X}_k(i) \leftarrow \text{RandomInitialization}(i)$
end for
$c = 0$ // Iteration counter: initialization
$e_{tot}^c = \text{inf}$ // Global error: initialization
repeat
  $c = c + 1$
  for each word $w_i^k \in \mathcal{V}_k$ **do**
    for each direction z **do**
      $\mathbf{X}_k(i) \leftarrow \text{MoveWordToDirection}(\mathbf{X}_k(i), z)$
      $\mathbf{e}(z) \leftarrow \text{ComputeLocalError}(\tilde{\mathbf{S}}_k, \mathbf{P}_k, \mathbf{X}_k, i)$
    end for
    $\hat{z} \leftarrow \text{FindDirectionOfMinLocalError}(\mathbf{e})$
    $\mathbf{X}_k(i) = \text{MoveWordToDirection}(\mathbf{X}_k(i), \hat{z})$
  end for
  $e_{tot}^c \leftarrow \text{UpdateGlobalError}(\tilde{\mathbf{S}}_k, \mathbf{P}_k, \mathbf{X}_k)$
until $e_{tot}^{c-1} < e_{tot}^c$ // Stopping condition
$\hat{\mathbf{P}}_k \leftarrow \text{SparseSimilarityReestimationNormalizedRanges}(\mathbf{X}_k, \tilde{\mathbf{S}}_k)$
return $\mathbf{X}_k$ // $n \times d$ matrix with coordinates;
return $\hat{\mathbf{P}}_k$ // $n \times n$ sparse-normalized similarities;

Sparse Similarity Re-estimation Normalized Ranges: Given the matrix $\mathbf{X}_k \in \mathbb{R}^{n \times d}$ containing the vector projections of words in the d-dimensional space, the dissimilarity matrix, $\mathbf{D}_k \in \mathbb{R}^{n \times n}$, is re-estimated using the Euclidean distance as defined in Eq. 3.5. For another research area, one could directly use the distances $\mathbf{D}_k$ either as is or normalized, but regarding the lexical space the embodiment of sparsity in proximities seems straight forward. Let d correspond to the maximum distance of connected word pairs:

$$d = \max_{i,j} \{\tilde{\mathbf{S}}_k(i,j) \cdot \mathbf{D}_k(i,j)\} \tag{3.8}$$

then $\hat{\mathbf{P}} \in \mathbb{R}^{n \times n}$ is simply the sparse distances normalized in [0,1] and transformed to similarities:

$$\hat{\mathbf{P}}_k(i,j) = \tilde{\mathbf{S}}_k(i,j) \cdot (1 - \frac{\mathbf{D}_k(i,j)}{d}) \tag{3.9}$$

In Fig. 3.6 an illustrative example is depicted from the projection of a small neighborhood of word 'book', using sparse percentage 70%. It is interesting to observe the inter-connected clusters that correspond to the two senses of 'book', i.e., *book as read* and *book as reservation* that are represented under the same space. Another interesting point is that although the low-dimensional representations of relatively dissimilar words 'publication' and 'reservation' is close enough, they are not 'connected' in the graph, so the representation of one word is 'unseen' to the other.

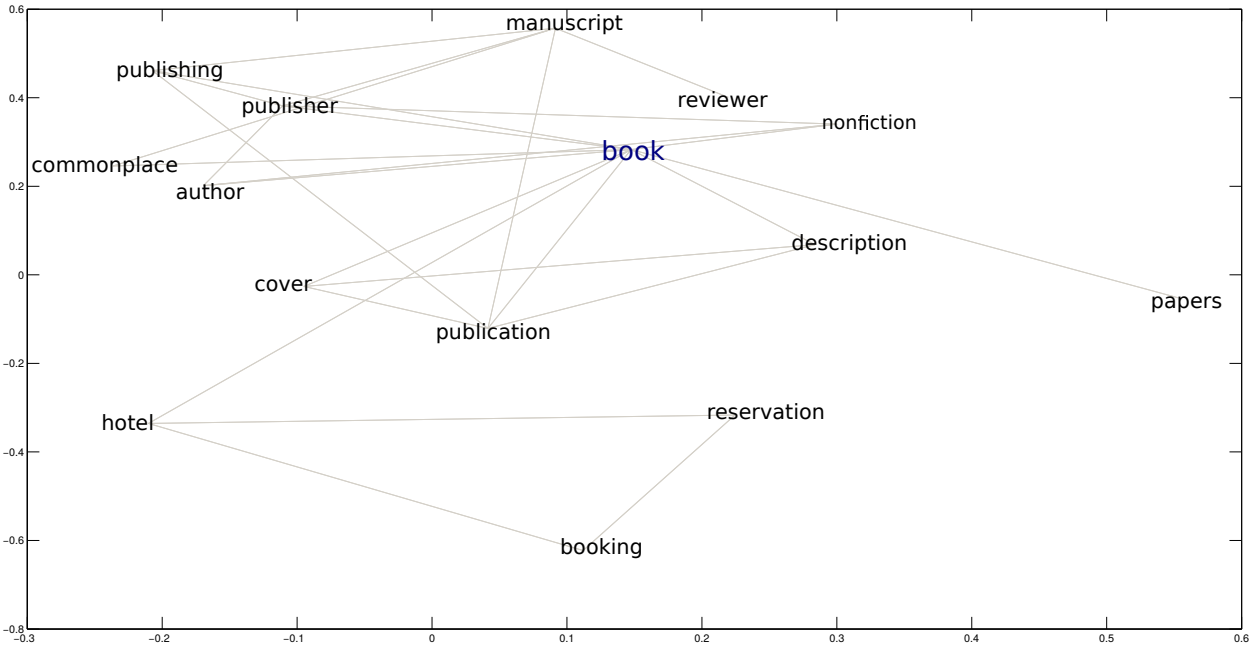

Figure 3.6: Example of projection of neighborhood of word 'book' using SP algorithm with sparsity 70% and projection to 2 dimensions.

3.2.4 Fusion from different subspaces

To reach a decision on the strength of the semantic relation between words w_i and w_j the sparse semantic similarity matrices from each domain $\hat{\mathbf{P}}_k$ must be combined. Only domains were both words w_i and w_j appear are relevant in this fusion process. This procedure is described next.

Motivation: Given a set of words $L = \{w_1, w_2, ...w_n\}$ we assume that their corresponding set of word senses[6] is $M = \{s_{11}, s_{12}, .., s_{1n_1}, ..., s_{n1}, s_{n2}, .., s_{nn_n}\}$. The set of senses is defined as $M = \cup_{i=1}^{n} M_i$, where $M_i = \{s_{i1}, s_{i2}, ..., s_{in_i}\}$ is the set of senses for word w_i. Let $S(.)$ be a metric of semantic similarity, e.g., the metric defined in Eq. 2.3, which is symmetric, i.e., $S(x, y) \equiv S(y, x)$. The notations $S_w(.)$ and $S_s(.)$ are used in order to distinguish the similarity at word and sense level, respectively. According to the maximum sense similarity assumption [83], the similarity between w_i and w_j, $S_w(w_i, w_j)$, is defined as the pairwise maximum similarity between their corresponding senses $S_s(s_{ik}, s_{jl})$:

$$S_w(w_i, w_j) \equiv S_s(s_{ik}, s_{jl}), \quad \text{where} \quad (k, l) = \underset{(p \in M_i, r \in M_j)}{\operatorname{argmax}} S_s(s_{ip}, s_{jr}) \tag{3.10}$$

Note that the maximum pairwise similarity metric (or equivalently the *minimum pairwise distance metric*) is also known as the "common sense" set similarity (or distance) employed by human cognition when evaluating the similarity (or distance) between two sets.

Fusion of local dissimilarity scores: Next we describe a manifold fusion model that follows the maximum pairwise similarity principle motivated by human cognition. The steps for the re-computation of the (global) similarity between words w_i and w_j are:

1. Search for all the manifolds where w_i and w_j co-exist.
2. Let $\mathcal{U}$ be the indexes of the subset of manifolds from the previous step. The similarities between words w_i and w_j are retrieved from domain similarity matrices $\hat{\mathbf{P}}_u$ for all $u \in \mathcal{U}$. The similarities are stored into vector $\mathbf{d} \in \mathbb{R}^{|\mathcal{U}| \times 1}$.
3. Motivated by the maximum sense similarity assumption (see above) the global similarity between w_i and w_j is defined as[7]:

$$\hat{\mathbf{P}}(i, j) = \max_{m=1..|\mathcal{U}|} \{\mathbf{d}_m\} \tag{3.11}$$

4. If words w_i and w_j do not co-exist in any domain then 0 is assigned as their similarity score.

For example, let one pair of words (w_1, w_2) co-exists in $|\mathcal{U}| = 3$ different domains with corresponding local similarities $\mathbf{d} = [0.43\ 0.67\ 0.55]$ then the global similarity (w_1, w_2) is

[6] This is a simplification. In reality, some of the word senses will be the same, so strictly speaking this is not a set definition.

[7] Other fusion methods have also been evaluated, e.g., (weighted) average. Maximum pairwise similarity fusion outperformed other fusion schemes.

0.67.

3.2.5 Extension of LDMS System

In this section an extension of LDMS System is presented that is depicted in Fig. 3.7.

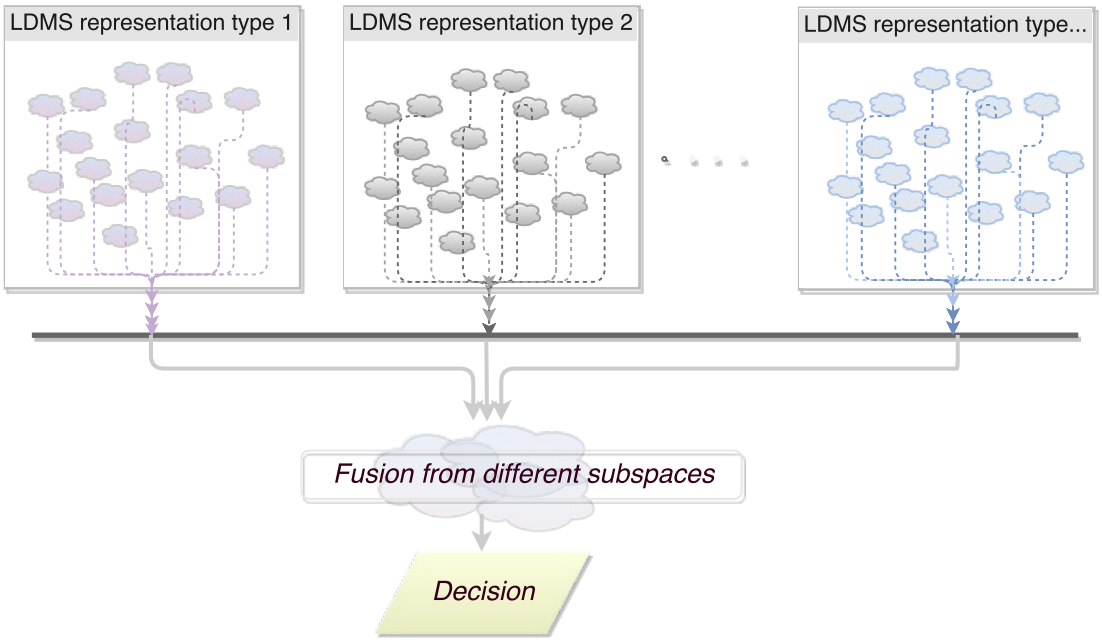

Figure 3.7: Extension of LDMS system.

The main purpose of this architecture is to allow multiple types of representations to co-exist in parallel. For example, in LDMS representation type 1 manifolds may be built based on semantic similarity relations among words, in LDMS representation type 2 manifolds may be built based on affective relations among words. Another example where the extension of LDMS system can be applied is the construction of manifolds of one representation type with the method 'Neighborhoods' and the construction of the second representation type with the method 'Hierarchical Categories' (both described in Sec. 3.2.1). Thereafter, in global level much different kind of relations can be fused in order to reach a global decision. Note that the fusion method is the same as described in Sec. 3.2.4 with the difference that the search of co-existance of a pair of words is performed to all different representation types.

Chapter 4

Evaluation

In this chapter, we describe the experimental procudure and the evaluation results; we present the methodology of corpus creation, the datasets that are utilized for evaluation purposes and we analyse all the evaluation tasks. First, we will describe the creation of the vocabulary set and its corresponding corpus, and then we will present two evaluation tasks that will be used in order to measure the performance of various DSMs approaches, namely: (i) similarity judgment between nouns (see Section 4.2), and (ii) creation of a taxonomy consisting of nouns (see Section 4.4).

4.1 Vocabulary, Corpus Description and Baseline Similarities

One English vocabulary was utilized for evaluation purposes, that contains mainly nouns of English language. The English nouns were extracted from the SemCor3 corpus [1] ,resulting a vocabulary set of 8752 words. This vocabulary was used for the creation of a big corpus, based on the methodology described in Sec. 2.2.1, resulting a corpus consisting of approximately 8752000 snippets, since 1000 snippets were acquired for each word of vocabulary.

Thereafter, based on this corpus, a set of baseline similarity metrics were applied for the computation of similarity between words of vocabulary. Namely, the 'Google-based semantic relatedness' described in Sec. 2.2.3, the 'Context-1' and 'Context-5' that correspond to the context based similarity computation method described in Sec. 2.2.2 using Eq. 2.1 as similarity metric and contextual window size 1 word and 5 words respectively and the 'Dice coefficient' described in Sec. 2.2.3.

[1] http://www.cse.unt.edu/rada/downloads.html

4.2 Similarity Judgment

Similarity judgment constitutes the first and most essential evaluation task, where the performance of similarity metrics is evaluated against human judgments. More specifically, there have been constructed a set of datasets containing word pairs, for example one pair may be the ("tiger", "animal"), and humans have rated the semantic similarity of those pairs based on their own judgment. Thus, the developed similarity metrics should be correlated as much as possible to the human judgments. Note that, when the data collection is not conducted in the controlled lab environment, e.g., data is collected via web forms or from web tasks such as crowd-sourcing, subjects tend to rate the degree of semantic relatedness rather than similarity. In essence, the 'natural' task that comes with little effort to humans is to rate associations rather than similarity. In order to rate similarity, subjects have to exert higher-cognitive (semantic) effort and be carefully instructed to do so. The datasets that will be used for evaluation have been developed in much diverse environments, so it is expected to observe diversity to the performance of each metric to the different datasets. Another issue of the diversity of the datasets is that the creation of each dataset may have been focused to a specific issue, for example to rare words. The description of the different evaluation datasets is presented bellow:

1. **WS353** [84]: WordSimilarity-353 is a widely used dataset for evaluation of semantic similarity and is a set of 353 English word pairs along with human-assigned similarity judgments. WS353 is a collection of pairs for measuring both word similarity and relatedness, so it has been splitted into two subsets, one for evaluating similarity, and the other for evaluating relatedness, according to the procedure described in [85], resulting to the two following datasets.

2. **WSsim** [85]: Subset of 202 pairs for evaluating similarity.

3. **WSrel** [85]: Subset of 252 pairs for evaluating relatedness.

4. **RG** [86]: Rubenstein and Goodenough (RG) is a set of 65 noun pairs with human similarity ratings, that is also widely used.

5. **MC** [87]: Miller and Charles (MC) is an also known set of 30 noun pairs with human similarity ratings.

6. **RW** [88]: Stanford Rare Word (RW) similarity dataset consists of 2034 pairs with the corresponding human judgments, that is computed as the average similarity rating

using up to 10 individuals.

7. **MEN** [89]: MEN is a recently developed dataset containing 3000 word pairs of randomly selected words from corpora, where the ground truth judgments were obtained based on crowd-sourcing.

8. **MTurk287** [90] The human ratings of MTurk-287 collection were obtained through the Amazon's Mechanical Turk workers, resulting in a set of 287 word pairs labeled overall. Up to 30 workers were assigned per pair, with an average of 23 MTurk workers rating each word pair.

9. **MTurk771** [91]: The Mturk-771 set was recently created and contains 771 English word pairs along with human-assigned relatedness judgments.

The Pearson's correlation coefficient was used as evaluation metric to compare estimated similarities against the ground truth. Let $\mathbf{x} = [x_1, x_2, ..., x_m]^T$ and $\mathbf{y} = [y_1, y_2, ..., y_m]^T$ be the vectors that contain the similarity scores given by human subjects and the computational metric, respectively, if m is the number of word pairs of the dataset then, Pearson correlation coefficient is defined as follows:

$$\rho_{\mathbf{xy}} = \frac{\sum_{i=1}^{m}(\mathbf{x}_i - \tilde{x})(\mathbf{y}_i - \tilde{y})}{\sqrt{\sum_{i=1}^{m}(\mathbf{x}_i - \tilde{x})^2 \sum_{i=1}^{m}(\mathbf{y}_i - \tilde{y})^2}} \tag{4.1}$$

where $\tilde{x}$ and $\tilde{y}$ are the sample means of $\mathbf{x}$ and $\mathbf{y}$, for $i = 1, 2, ..., m$. This coefficient was selected instead of Spearman's rank correlation coefficient to retain the initial scaling of similarities in the evaluation metric, as opposed to the alternation of this scaling through the transformation of similarities into ranks.

4.2.1 Performance of Various DSMs

For this evaluation experiment the lexicon of 8752 nouns is utilized, that is described in Sec. 4.1. Thus, all the evaluated datasets are filtered according to the lexicon, resulting 280 pairs for WS353, 156 pairs for WSsim, 206 for WSrel, 57 for RG, 28 for MC, 151 for RW, 1477 for MEN, 126 for MTurk287 and 592 for MTurk771. In Table 4.1 the performance of some well known *unsupervised* similarity estimation algorithms is reported as well as the performance of the proposed LDMS system, more specifically:

1. Google Rel, correspond to the co-occurrence metric 'Google-based semantic relatedness' described in Sec. 2.2.3.

2. Context-1 correspond to the context based method described in Sec. 2.2.2 using Eq. 2.1 as similarity metric and contextual window size 1 word.

3. Context-5 is the same with previous method with contextual window size 5 words,

4. Dice, correspond to the co-occurrence metric 'Dice coefficient' described in Sec. 2.2.3.

5. Word2vec [35, 92, 93], tool was used. We applied the CBOW approach (as being more computationally efficient) for context window 3. The dimensions of the resulting words-features matrix was set to 300.

6. WikiRelate! includes various taxonomy-based metrics that are typically applied to the WordNet hierarchy; the basic idea behind WikiRelate! is to adapt these metrics to hierarchy extracted from the links between the pages of English Wikipedia [94].

7. TypeDM is a well known structured DSM [15], i.e., employs syntactic information from corpus.

8. AAHKPS1 constitutes an unstructured paradigm of DSM development using four billion web documents that were acquired via crawling [85]

9. SEMNET is the alternative implementation of unstructured DSMs based on the idea of semantic neighborhoods and networks, described in Sec. 2.2.4 and the reported results are for neighborhood size equal to 100.

10. LDMS is the manifold based distributional semantic model proposed in this work. The reported performance correspond to the following parameters set, construction of manifolds: 160 neighbors of each word (descriped in 3.2.1), sparsity method 'triangle inequality target word' (descriped in Sec. 3.2.2) and projection dimension 5 dimensions.

Algorithm	**Datasets**								
	WS353	WSsim	WSrel	RG	MC	RW	MEN	MTurk287	MTurk771
Google Rel	0.61	0.65	0.64	0.81	0.85	0.46	0.74	0.61	0.61
Context-1	0.30	0.34	0.27	0.52	0.51	0.21	0.46	0.50	0.39
Context-5	0.13	0.01	0.18	0.29	0.21	0.10	0.25	0.42	0.20
Dice	0.22	0.23	0.32	0.60	0.60	0.25	0.45	0.36	0.39
Word2Vec	0.58	0.57	0.67	0.79	0.79	**0.48**	0.72	0.66	0.62
WikiRelate!	0.48	-	-	0.53	0.45	-	-	-	-
TypeDM	-	-	-	0.82	-	-	-	-	-
AAHKPS1	-	-	-	-	0.89	-	-	-	-
SEMNET	0.64	-	-	**0.87**	0.91	-	-	-	-
LDMS	**0.73**	**0.75**	**0.74**	0.84	**0.95**	**0.48**	**0.74**	**0.73**	**0.67**

Table 4.1: Performance of different DSMs to various datasets for the task of similarity judgment.

The metrics Google Rel, Context-1, Context-5, Dice, Word2Vec are computed based on the corpus of 8752 nouns described in Sec. 4.1. The performance of the other metrics is retrieved from the corresponding papers, thus their performance is not reported for all datasets.

In the task of semantic similarity estimation against human judgments, baseline metric 'Google-based semantic relatedness' perform well to all datasets, compared to the other baseline metrics, Context-1, Context-5 and Dice. The AAHKPS1 also reported good performance for the MC dataset, this can be attributed to the fact that too many documents were analyzed in the specific DSM. The network based method SEMNET, performed pretty well to all datasets, this can be attributed to the fact that in SEMNET a more sophisticated representation of lexical space is utilized and the similarities are computed on the top of those representations. The LDMS approach outperformed all the reported (and unsupervised) DSMs in all datasets, except the RG dataset, where the performance is close enough to the best reported performance, i.e. 0.84 vs 0.87, but as we will show subsequently, 0.87 correlation is also achieved from LDMS using another parameter set.

4.2.2 LDMS System: Analysis of Different Scenarios

In this section, we analyze the performance of LDMS approach to different scenarios for similarity judgment task, more specifically, we will evaluate two different methods for the construction of manifolds. In Sec. 4.2.2, the manifolds are built as the neighborhoods of words in vocabulary, while in section Sec. 4.2.2 we present an attempt of addition of manifolds with categorical relations along with the neighborhood-manifolds. The similarities computed by Google-based Semantic Relatedness (defined by Eq. (2.2)) are used as baseline and as bootstrap similarity measures of LDMS system.

Construction of Manifolds: Semantic Neighborhoods

The performance (Pearson correlation) of the LDMS approach is presented, as a function of neighborhood size and sparseness method (with fixed 5 projection dimensions), in the following figures: Fig. 4.1 for WS353 and RG datasets, Fig. 4.2 for MC and MEN datasets and Fig. 4.3 for MTurk287 and MTurk771 datasets. The baseline performance is also plotted (dotted line) and noted as 'Base'. The analyzed sparseness methods, described in Sec. 3.2.2, are the following: 1) 'percentage' where a degree of sparseness is fixed and used to all manifolds, here we will present the degrees 0%, 80% and 90% and 2) 'triangle inequality target word', where the sparseness of each manifold is defined as function of the metric properties of the manifold, noted as 'trInTar'.

For all six datasets, we see a clear relationship between neighborhood size, sparseness method and performance. Sparse representations achieve peak performance for larger neighborhood sizes. High degree of sparseness, between 80% and 90%, achieves high results for manifold/neighborhood sizes between 100 and 200. The abrogation of sparseness, i.e. 0% degree of sparseness or equivalently the usage of MDS algorithm for the projection of manifolds, performs poorly to all datasets, this is a strong indication of the importance of sparseness property to representations dealing with the modeling of lexical space. Another interesting point is that, the sparse method 'triangle inequality target word', which constitutes a more sophisticated approach, is the best performing method. The low performance of LDMS using small neighborhood sizes, is somewhat expected because as the neighborhood size decreases, so the probability of a random word pair to co-exist in a manifold decreases. Thus too many pairs of the datasets are considered as 'sparse', i.e., with 0 corresponding similarity, when small neighborhoods are used. Finally, it is clear that LDMS outperformed the baseline metric in all datasets.

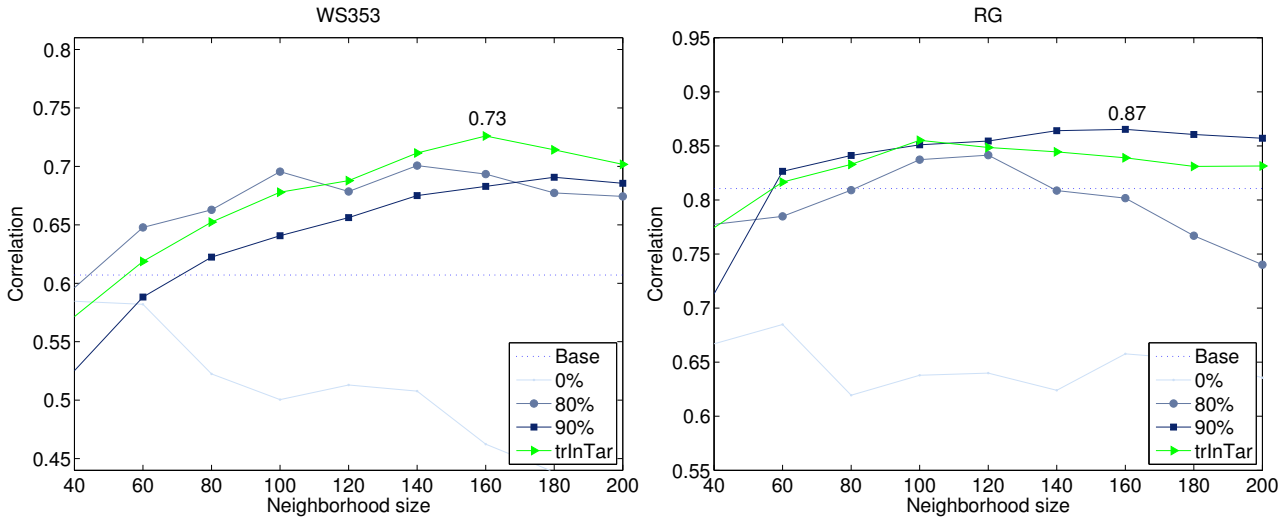

Figure 4.1: Performance as a function of manifold size, n, and sparseness method for the (a) WS353 dataset and (b) RG dataset.

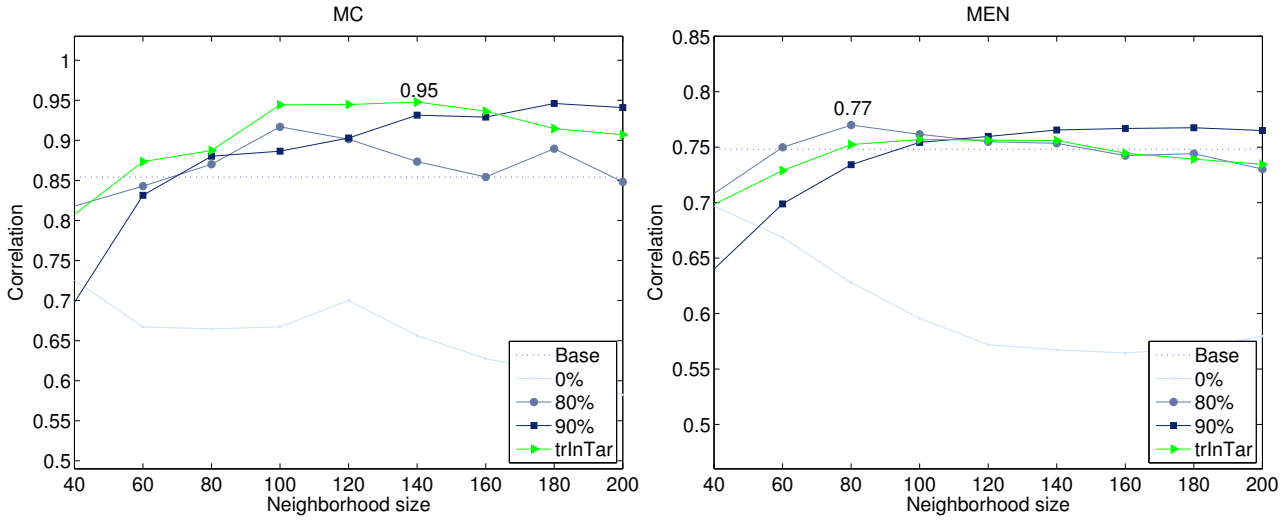

Figure 4.2: Performance as a function of manifold size, n, and sparseness method for the (a) MC dataset and (b) MEN dataset.

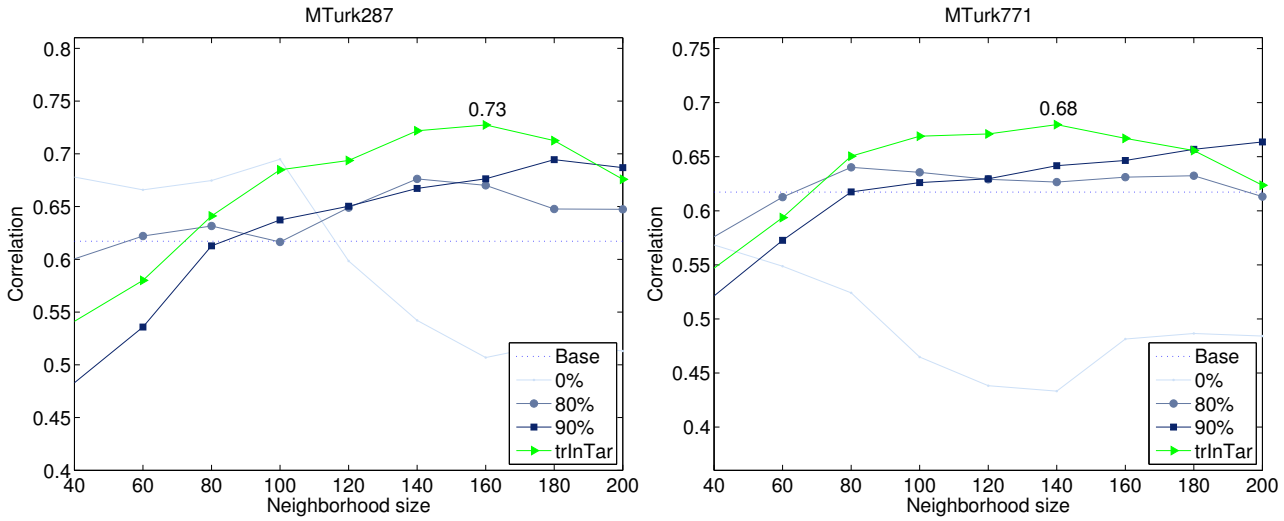

Figure 4.3: Performance as a function of manifold size, n, and sparseness method for the (a) MTurk287 dataset and (b) MTurk771 dataset.

In Fig. 4.4 we present the performance of LDMS system to WSsim and WSrel datasets. These datasets constitute the categorization of pairs of WS353 dataset to pairs with similarity connections and pairs with relatedness connections. The performance is presented as a function of manifold size, n, and sparseness method, with 5 projection dimensions.

It is interesting to observe that LDMS system manage to capture both similarity and relatedness connections and improve significantly the baseline. Neighborhood sizes between 140 and 180 along with high sparse percentages lead to high performance. Once again, the peak performance is achieved through the sparse method 'triangle inequality target word' for both datasets; while 0% sparsity, leads to very low performance.

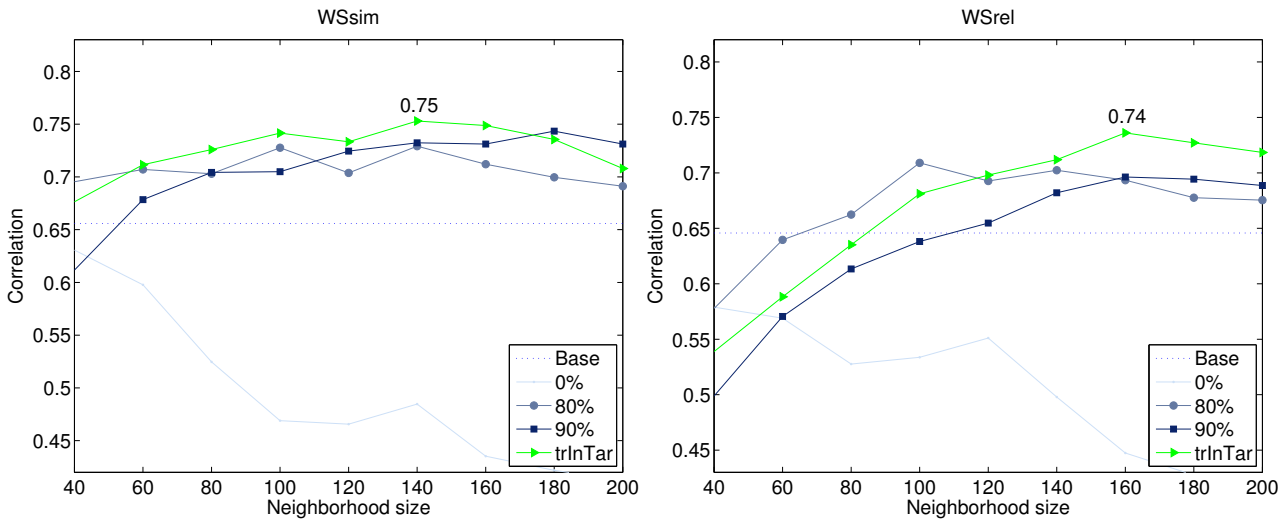

Figure 4.4: Categorization of dataset pairs based on similarity and relatedness connections: Performance as a function of manifold size, n, and sparseness method for the (a) WSsim dataset evaluating similarity and (b) WSsim dataset evaluating relatedness.

In Fig. 4.5 we present the performance of LDMS system to WS353 and MTurk287 datasets, where the pairs of each dataset have been categorized based on their abstraction level. More specifically, following the methodology described in [79], we assigned a concreteness score, ranging in [-1,1], to all 8752 words. Thus, words with corresponding concreteness score in [0,1] are considered concrete, while words with concreteness score in [-1,0) are considered abstract. Thereafter, we were able to categorize the pairs of each dataset, to abstract-abstract pairs, abstract-concrete pairs and concrete-concrete pairs and create 3 corresponding subsets of the dataset. The performance of WS353 and MTurk287 datsets along with their corresponding subsets is presented as a function of manifold size, n, using as sparseness method the 'triangle inequality target word' and 5 projection dimensions.

LDMS system manage to capture very well the relations of abstract-concrete pairs, since the best performing subset is the abstract-concrete for both datasets. An overall observation is that the subsets that contain concrete words, i.e., abstract-concrete and concrete-concrete, achieve high performance for both cases. On the other hand, LDMS performs poorly for abstract-abstract pairs. The peak performance is achieved for neighborhood sizes between 100 and 200 for all pairs of each dataset and their corresponding subsets.

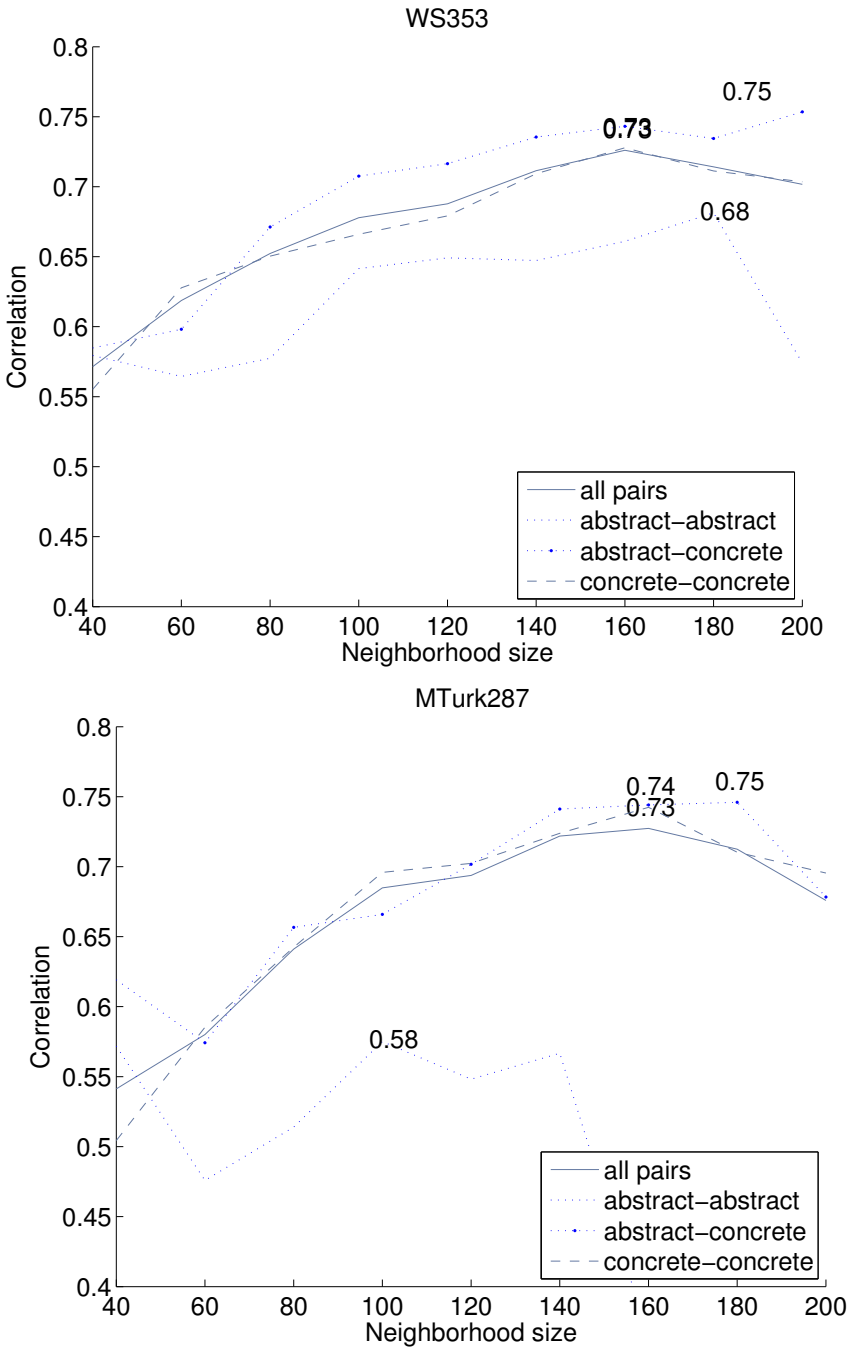

Figure 4.5: Categorization of dataset pairs based on abstraction: Performance as a function of manifold size, n, with sparseness method *'Triangle inequality target word'*, for the (a) WS353 dataset (b) MTurk287 dataset.

The performance of LDMS is shown in Fig. 4.6 as a function of the projection dimension d,

for WS353, WSsim, WSrel, RG, MC, MEN, MTurk287 and MTurk771 datasets. The method of sparseness is the 'triangle inequality target word' while the manifold/neighborhood size equals to 160 for all experiments.

It is interesting to observe that the performance of LDMS system remains relatively constant when at least $d = 3$ is used, for all eight datasets . The results suggest that even *a 3D sub-space is adequate for accurately representing the semantics of each underlying manifold.*

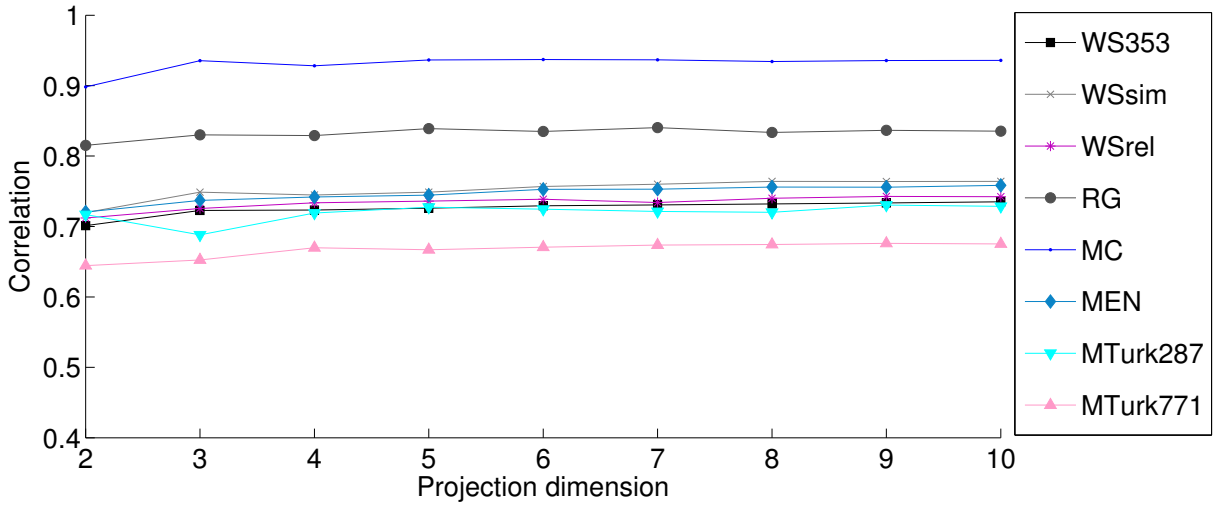

Figure 4.6: Performance of the WS353, WSsim, WSrel, RG, MC, MEN, MTurk287 and MTurk771 datasets as a function of projection dimension d.

Construction of Manifolds: Semantic Neighborhoods and Hierarchal Categories

Here, we will present the performance of extended LDMS system, described in Sec. 3.2.5. More specifically, additively to the manifolds build from semantic neighborhoods of each word we have added manifolds that are build based on hierarchical relations among words, as described in Sec. 3.2.1. To built the manifolds based on hierarchical relations we have utilized the hierarchical tree structure of WordNet.

For each word of 8752 vocabulary nouns, we have marked the categories, the super-categories, the super-super-categories, etc., assigned to the word, until we reach the root category of WordNet, i.e., the category 'entity' that is root of every vocabulary word. Thus,

we extracted a set of 5800 different categories from all vocabulary words and the information of which words are assigned to each one of the 5800 categories. Then, for each category we built one manifold containing the words that are assigned to the category, resulting a new set of 5800 manifolds.

Since, these categories belong to different hierarchical levels, the size of manifolds that correspond to higher level of hierarchy will contain much more words than the manifolds that correspond to lower levels of hierarchy, for example the manifold that correspond to category 'entity' will contain all the vocabulary words. Thus, we threw the top 60 manifolds that correspond to categories of top level of hierarchy because their size was too large and whole idea of LDMS system is the construction of relatively small manifolds containing words with strong relations. Also, we threw manifolds of too low level of hierarchy containing less than 3 words. Thus, the new set contained in total 2917 manifolds. In Fig. 4.7 the distribution of size of manifolds built from hierarchical relations is presented. The size of almost 80 manifolds range between 100 and 240, while the size of almost 210 manifolds range between 40 and 100 and the size of most manifolds is less than 40.

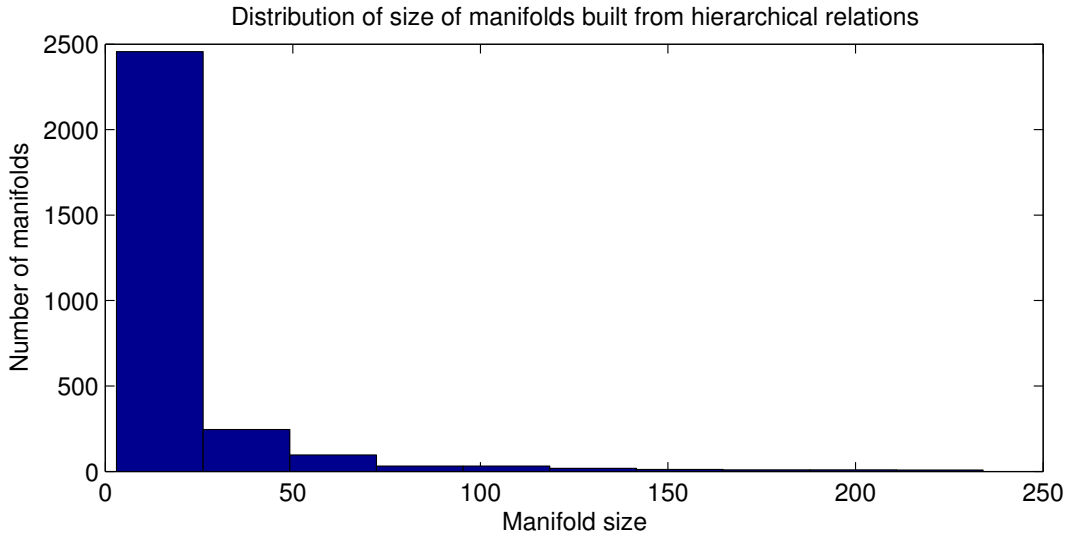

Figure 4.7: Distribution of size of manifolds built from hierarchical relations.

The new manifolds built from hierarchical relations are utilized from LDMS system additively with the manifolds built from semantic neighborhoods. In Figs. 4.8, 4.9 the performance of WS353, MTurk287, WSsim and WSrel datasets is presented as function of neighborhood size and sparseness method. Note that, while the manifolds built from neighborhoods change as function to neighborhood size, the manifolds built from hierarchal

relations are the same for all experiments. Since size of most manifolds built from hierarchal relations are very small, there is no need for extra sparseness, so the sparse method that will be utilized for the hierarchical representation is the 'Similarity 0' described in Sec. 3.2.2. The sparse methods that are utilized to manifolds built from neighborhoods are the percentage '0%', '80%', '90%' and the triangle inequality target word, described in Sec. 3.2.2. The baseline metric, reported as 'Base', is the 'Google-based semantic relatedness'.

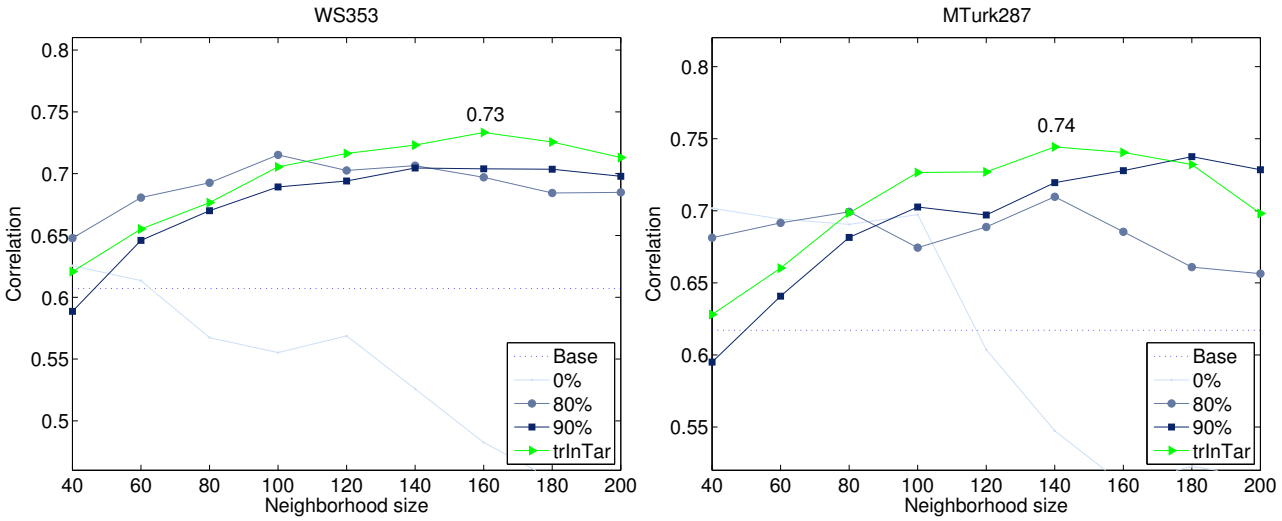

Figure 4.8: Performance as a function of manifold size, n, and sparseness method for the (a) WS353 dataset and (b) MTurk287 dataset.

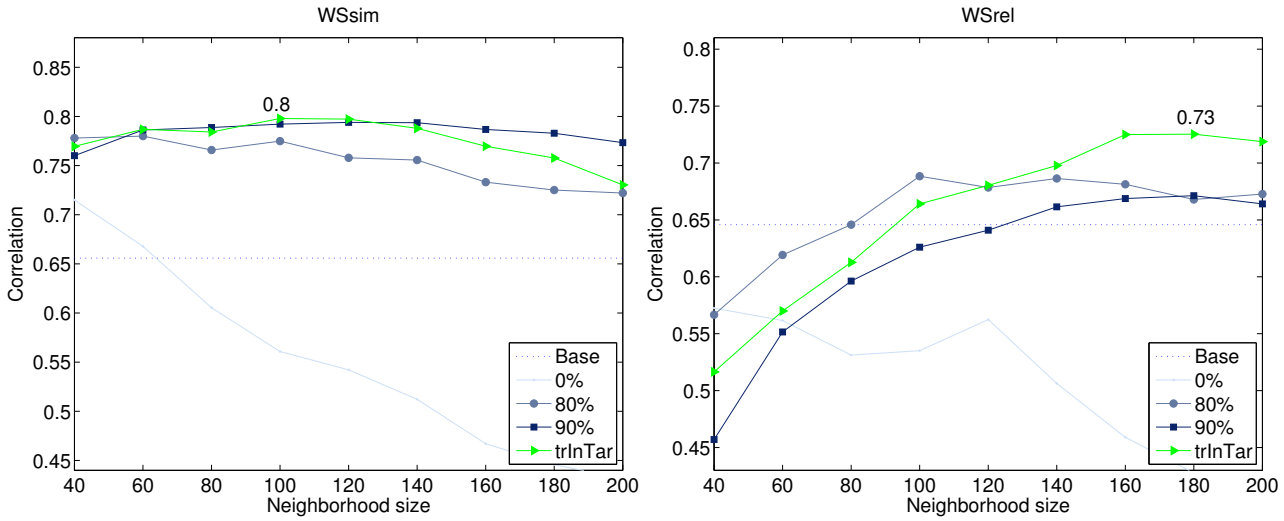

Figure 4.9: Performance as a function of manifold size, n, and sparseness method for the (a) WSsim dataset evaluating similarity and (b) WSsim dataset evaluating relatedness.

The performance of WS353, MTurk287 and WSrel datasets have not changed (not for better or for worst) regarding the approach where only the representation of neighborhoods was used. But, a very interesting finding is that the performance of WSsim has been considerably improved when the hierarchal relations have been utilized. More specifically the best performance has increased from 0.75 to 0.80 when hierarchal relations have been utilized. This, is a strong indication that more sophisticated hierarchal representations are needed in order to capture the semantic similarity relations among words, rather than just the relatedness.

4.3 Comparison With Other Representation Algorithms

Two well-established dimensionality reduction algorithms (Isomap and LLE) that support the manifold hypothesis, were applied to the task of semantic similarity computation and their performance is presented in Table 4.2. Note that, LDMS is not directly comparable with Isomap-LLE algorithms because it represents only the manifolds in low-dimensional spaces and not the whole dataset. LDMS, Isomap and LLE were given as input the global similarity matrix 'Google-based semantic relatedness' of 8752 words. Isomap and LLE used dimensionality reduction down to $d = 5$ and neighborhood size equal to 120. While

LDMS run for dimensionality down to $d = 5$, manifold/neighborhood size equal to 160 and sparseness method 'triangle inequality target word', described in Sec. 3.2.2.

Datasets	Algorithm			
	Baseline	Isomap	LLE	**LDMS**
WS353	0.61	0.14	0.04	0.73
RG	0.81	0.04	0	0.84
MC	0.85	-0.04	-0.04	0.95

Table 4.2: Performance Isomap and LLE representation algorithms for the task of similarity judgment for WS353, RG and MC datasets.

The proposed LDMS system surpassed the performance of the other systems for all three datasets. Both Isomap and LLE dimensionality reduction algorithms are shown to perform poorly for this particular task. The poor performance of Isomap and LLE can be attributed to the nature of the specific application, i.e., word semantics. A key characteristic of this application is the ambiguity of word senses. These algorithms assume only one sense for each word (i.e., a word is represented as a single point in a high-dimensional space). Although the disambiguation task is not explicitly addressed, LDMS approach handles the ambiguity of words by isolating each word's senses in different manifolds.

4.4 Taxonomy Creation

In this section, the evaluation task of the creation of simple taxonomy of nouns is presented. In particular, the ESSLLI dataset [1] was used, which constitutes a three-level taxonomy depicted in Figure 4.10.

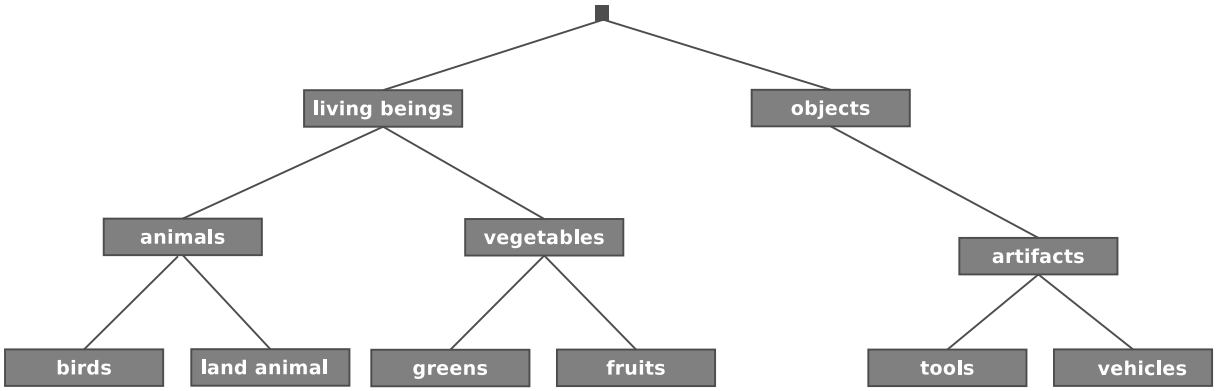

Figure 4.10: Taxonomy of ESSLLI dataset [1]

The lowest level of the taxonomy consists of (instances of) the following six concepts: (i) "birds", (ii) "land animals", (iii) "greens", (iv) "fruits", (v) "tools", and (vi) "vehicles". The middle level includes the concepts (i) "animals", (ii) "vegetables", and (iii) "artifacts", while the upper level is distinguished into "living beings" and "objects".

The original ESSLLI dataset consists of 44 nouns (instances). Given the similarity matrix between those nouns, for each taxonomic level the K-means clustering algorithm is incorporated, in order to separate the nouns to different clusters. The purity of clusters, r, was used as evaluation metric, defined as [15]:

$$r = \frac{1}{c} \sum_{i=1}^{K} \max_{j}(c_i^j), \tag{4.2}$$

where c_i^j is the number of nouns assigned to the i^{th} cluster that belong to the j^{th} groundtruth class. The number of clusters is denoted by K, while c is the total number of nouns included in the dataset. Purity expresses the fraction of nouns that belong to the true class, which is most represented in the cluster [15], taking values in the range $[0, 1]$, where 1 stands for perfect clustering.

4.4.1 Performance of Various DSMs

For this evaluation experiment the lexicon of 8752 nouns will be utilized, described in Sec. 4.1. We used a subset of ESSLLI dataset's nouns that was covered by the lexicon of 8752 nouns, resulting 31 nouns instances. The performance of the proposed LDMS approach (dimensionality $d = 5$, manifold/neighborhood size equal to 160 and sparseness

method 'triangle inequality target word', described in Sec. 3.2.2) vs the baseline, as well as of other similarity estimation algorithms, that are described in Sec. 4.4, is presented in Table 4.3

Algorithm	Top	Middle	Low
Google Rel	0.68	0.55	0.48
Context-1	0.58	0.55	0.48
Context-5	0.61	**0.58**	0.48
Dice	0.55	0.55	0.55
TypeDM	0.61	**0.58**	0.54
LDMS	**0.71**	**0.58**	**0.77**

Table 4.3: Performance of various algorithms to the taxonomy task for all three levels (top - middle - low).

LDMS yields better results than the other baseline metrics for the top and low taxonomic levels. Interestingly, for the low level of the taxonomy where results are rather poor for the baseline system, LDMS is shown to perform the best (better than the middle and top levels of the taxonomy).

Chapter 5

Conclusions

5.1 Conclusions

In this work, we proposed a novel, hierarchical DSM that was applied to semantic relation estimation task obtaining very good results. The proposed representation consists of low-dimensional manifolds operating in parallel that are derived from sparse projections of semantic neighborhoods. The core idea of low dimensional subspaces was motivated by cognitive models of conceptual spaces.

More specifically, the proposed system is composed from the following parts: 1. Construction of manifolds: this step constitutes the identification of manifolds, where each manifold is a set of items/words connected to each other with some kind of relation. In this thesis we experimented with two different methodologies for the construction of manifolds, the semantic neighborhoods of words and the manifolds created from hierarchal relations among words. 2. Sparse encoding of manifolds: this step is deals with the automatic construction of a sparse connectivity graph among items/words in each manifold, where the connection of two words in a graph indicates that those words are connected with some kind of relation, for example similarity relation. In this thesis we experimented with different methodologies for the construction of sparse connectivity mapping among words. 3. Low dimensional representation of manifolds: in this step each manifold is projected into a low-dimensional sub-space. Here, we proposed a dimensionality reduction algorithm, that constitutes an alternation of MDS, which that encounters the sparse connectivity matrices in order to perform the projection of the manifolds. 4. Fusion from different subspaces: Global operations are decomposed into local operations in multiple sub-spaces; results from these local operations are fused to come up with semantic relatedness estimates. Manifold DSM are constructed starting from a pairwise word-level semantic similarity matrix.

The validity of this motivation was experimentally verified via the estimation of semantic similarity between nouns. The proposed approach was found to be (at least) competitive with other state-of-the-art DSM approaches that adopt flat feature representations and do not explicitly include the manifolds, the sparsity and the dimensionality as key design

parameters.

Two other representation algorithms, Isomap and LLE, that respect the manifold hypothesis, were evaluated in the task of semantic similarity estimation. For each algorithm, firstly, a global representation of lexical space was constructed and thereafter semantic similarity estimations between word pairs were extracted based on this representation. The poor performance of Isomap and LLE can be attributed to the fact that one sense is assumed for each word (i.e., a word is represented as a single point in a high-dimensional space) and that sparsity is not a design system parameter. Hence, these algorithms are able to handle only one of from the three properties of lexical space. LDMS approach implicitly handles the ambiguity of words by isolating each word's senses in different manifolds.

Our initial intuition regarding the semantic fragmentation of lexical neighborhoods due to singularities introduced by word senses was supported by the high performance when large (i.e., 80% - 90%) degree of sparseness was imposed. The hypothesis of low-dimensional representation was validated by the finding that as little as three dimensions are adequate for representing domain/neighborhood semantics. It was also observed that the parameters of the LDMS model, i.e., number of dimensions, neighborhood size and degree of sparseness, are interrelated: very sparse projections achieve best results with very low dimensionality when large neighborhood sizes are used.

Also another property that proved to be a relevant and useful tool for the representation of lexical spaces was the triangle inequality. The triangle inequality was utilized as a method for building sparse connectivity matrices of the low dimensional spaces and this method produced the best results of our model for most evaluated datasets.

Afterwards, noun pairs have been categorized based on their abstraction level and the performance of abstract-abstract, abstract-concrete and concrete-concrete pairs was evaluated. LDMS system manage to capture very well the relations of abstract-concrete pairs, also an overall observation is that the subsets that contain concrete words, i.e., abstract-concrete and concrete-concrete, achieve high performance for both cases. On the other hand, LDMS performs poorly for abstract-abstract pairs.

Another interesting finding was that semantic similarity relations were considerably boosted when hierarchal relations were included in the system. Contrary semantic relatedness relations were not influenced. This indicates that more sophisticated hierarchal representations are probably needed in order to capture the semantic similarity relations among words.

5.2 Future Work

This is only a first step toward using ensembles of low-dimensional DSMs for semantic relation estimation. As future work we plan to develop algorithms designed to perform a more adaptive fragmentation of semantic space, i.e., further investigation of the creation of manifolds based on more complex geometric properties of the underlying space [95] is needed. The creation of multi-level hierarchical representations that are consistent with cognitive organization is an important challenge that can further improve manifold DSM performance.

A first approach of induction of hierarchical relations in the model has already been introduced that better captured the semantic similarity relations, hence the creation of hierarchal representations constitutes a very interesting research area for DSMs. Of course, another area that is very interesting for further investigation is the fusion from the different subspaces, especially when the hierarchical multi-level representations will be included in the system.

Currently, the proposed representation has been used only for the 'decision' of re-estimating the semantic similarity among words, proving to be very competitive from other DSMs. Additionally we could develop and other fusion schemes aiming to much different tasks (and not the similarity computation), such as the word sense disambiguation.

Finally, it would be very interesting to apply the proposed system to other languages and to various NLP applications, such as language modeling.

Appendix A

Definitions

In this chapter we present some useful definitions for this thesis. For a detailed analysis the reader is referred to [95].

A.1 Metric Space

Definition 1 (**Metric Space**)**.** Let $\mathcal{M}$ be a set and $d(.)$ be a distance function defined on $\mathcal{M} \times \mathcal{M}$.[1] A metric space is a pair of $(\mathcal{M}, d)$ such that for all $\xi_1, \xi_2, \xi_3 \in \mathcal{M}$ we have:

1. d is real-valued, finite and nonnegative
2. $\xi_1 = \xi_2$ if and only if $d(\xi_1, \xi_2) = 0$
3. $d(\xi_1, \xi_2) = d(\xi_2, \xi_1)$ (Symmetry)
4. $d(\xi_1, \xi_2) \leq d(\xi_1, \xi_3) + d(\xi_3, \xi_2)$ (Triangle Inequality)

and it is defined as $(\mathcal{M}, d)$.

▼

Geometrically the triangle inequality states that the sum of lengths of any two sides of a triangle must be greater than the length of the third side, as depicted in the figure below.

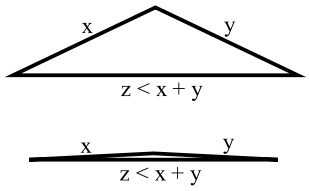

Figure A.1: Triangle inequality.

[1] The symbol $\times$ indicates the *Cartesian product* of sets.

A.2 Ball and Sphere

Definition 2 (**Ball and Sphere**)**.** Let $(\mathcal{M}, d)$ be a metric space, $\xi_0 \in \mathcal{M}$ and r be a real number where $r > 0$. Below we define three type of sets:

1. $\mathcal{B}(\xi_0; r) \triangleq \{\xi \in \mathcal{M} \ : \ d(\xi, \xi_0) < r\}$ (Open ball)
2. $\overline{\mathcal{B}}(\xi_0; r) \triangleq \{\xi \in \mathcal{M} \ : \ d(\xi, \xi_0) \leq r\}$ (Closed ball)
3. $\mathcal{S}(\xi_0; r) \triangleq \{\xi \in \mathcal{M} \ : \ d(\xi, \xi_0) = r\} = \overline{B}(\xi_0; r) \backslash B(\xi_0; r)$ (Sphere)

▼

A.3 Neighborhood

Definition 3 (**Neighborhood**)**.** Let $(\mathcal{M}, d)$ be a metric space and $\xi_0 \in \mathcal{M}$, the open ball $\mathcal{B}(\xi_0; \epsilon)$ is often called an ϵ-neighborhood of ξ_0. Neighborhood of ξ_0 is any subset of $\mathcal{M}$ which contains an ϵ-neighborhood of ξ_0.

▼

A.4 Power Set

Definition 4 (**Power Set**)**.** Let $\mathcal{M}$ be any set. The power set, $\mathcal{P}(\mathcal{M})$, of $\mathcal{M}$ is the set of all subsets of $\mathcal{M}$ including the empty set and $\mathcal{M}$ itself, i.e.,

$$\mathcal{P}(\mathcal{M}) \triangleq \{\mathcal{A} : \mathcal{A} \subseteq \mathcal{M}\} \tag{A.1}$$

▼

Bibliography

[1] M. Baroni, S. Evert, and A. Lenci, "Bridging the gap between semantic theory and computational simulations," in *In Proc. of ESSLLI Distributional Semantic Workshop*, 2008.

[2] S. S. Marmaridou, *Pragmatic meaning and cognition*. John Benjamins Publishing, 2000, vol. 72.

[3] A. Friedman and M. Thellefsen, "Concept theory and semiotics in knowledge organization," *Journal of documentation*, pp. 644–674, 2011.

[4] 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.

[5] Z. Estes, S. Golonka, and L. L. Jones, "8 thematic thinking: The apprehension and consequences of thematic relations," *Psychology of Learning and Motivation-Advances in Research and Theory*, p. 249, 2011.

[6] B. W. Whittlesea, "False memory and the discrepancy-attribution hypothesis: the prototype-familiarity illusion." *Journal of Experimental Psychology: General*, p. 96, 2002.

[7] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, pp. 467–479, 1992.

[8] N. Malandrakis, A. Potamianos, E. Iosif, and S. S. Narayanan, "Kernel models for affective lexicon creation." in *INTERSPEECH*, 2011, pp. 2977–2980.

[9] O. Corby, R. Dieng-Kuntz, F. Gandon, and C. Faron-Zucker, "Searching the semantic web: Approximate query processing based on ontologies," *Intelligent Systems, IEEE*, vol. 21, no. 1, pp. 20–27, 2006.

[10] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *arXiv preprint arXiv:1105.5444*, 2011.

[11] 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.

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

[13] 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.

[14] P. Gardenfors, "Conceptual spaces: The geometry of thought," *Cambridge, Massachusetts: USA. ISBN*, vol. 262071991, 2000.

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

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

[17] E. Iosif and A. Potamianos, "Similarity computation using semantic networks created from web-harvested data," Natural Language Engineering (DOI: 10.1017/S1351324913000144), 2013.

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

[19] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[20] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[21] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita, "Dimensionality reduction using non-negative matrix factorization for information retrieval," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 2, 2001, pp. 960–965 vol.2.

[22] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[23] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[24] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Foundations of computational mathematics*, vol. 9, no. 1, pp. 51–77, 2009.

[25] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 287–296.

[26] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[27] J. Wang, "Maximum variance unfolding," in *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Springer, 2011, pp. 181–202.

[28] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 639–655.

[29] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Advances in Neural Information Processing Systems*, 2009, pp. 2223–2231.

[30] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2013.

[31] A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of semantic distance," *Computational Linguistics*, vol. 32, pp. 13–47, 2006.

[32] E. Iosif, "Network-based distributional semantic models," Ph.D. dissertation, ECE Department, Technical University of Crete, Kounoupidiana, Chania, 2013.

[33] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Advances in neural information processing systems*, 2001.

[34] P. D. Turney, "Similarity of semantic relations," *Computational Linguistics*, pp. 379–416, 2006.

[35] I. Mikolov, T.and Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[36] O. Barkan, "Bayesian neural word embedding," *arXiv preprint arXiv:1603.06571*, 2016.

[37] O. Levy, Y. Goldberg, and I. Ramat-Gan, "Linguistic regularities in sparse and explicit word representations." in *CoNLL*, 2014, pp. 171–180.

[38] R. Lebret and R. Collobert, "Word emdeddings through hellinger pca," *arXiv preprint arXiv:1312.5542*, 2013.

[39] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2177–2185.

[40] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[41] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in neural information processing systems*, 2009, pp. 1081–1088.

[42] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH*, vol. 2, 2010, p. 3.

[43] L. Wittgenstein and G. E. M. Anscombe, *Philosophical investigations*. Blackwell Oxford, 1958, vol. 255.

[44] P. D. Turney, P. Pantel *et al.*, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, no. 1, pp. 141–188, 2010.

[45] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars." in *ACL (1)*, 2013, pp. 455–465.

[46] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 2013, p. 1642.

[47] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, 2013, pp. 746–751.

[48] E. Iosif and A. Potamianos, "Unsupervised semantic similarity computation between terms using web documents," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 11, pp. 1637–1647, 2010.

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

[50] S. Gauch and J. Wang, "A corpus analysis approach for automatic query expansion," in *Proceedings of the sixth international conference on Information and knowledge management*, 1997, pp. 278–284.

[51] E. M. Voorhees, "Query expansion using lexical-semantic relations," in *SIGIR'94*, 1994, pp. 61–69.

[52] E. Iosif and A. Potamianos, "Unsupervised semantic similarity computation using web search engines," in *Web Intelligence, IEEE/WIC/ACM International Conference*, 2007, pp. 381–387.

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

[54] 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.

[55] 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.

[56] T. Polajnar and S. Clark, "Improving distributional semantic vectors through context selection and normalisation," in *Proceedings of EACL*, 2014, pp. 230–238.

[57] 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.

[58] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[59] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2000.

[60] I. Borg, *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[61] S. Hassan and R. Mihalcea, "Semantic relatedness using salient semantic analysis." in *AAAI*, 2011.

[62] H. M. Meng and K. Siu, "Semiautomatic acquisition of semantic structures for understanding domain-specific natural language queries," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 14, no. 1, pp. 172–181, 2002.

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

[64] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proc. of International Conference on World Wide Web*, 2007, pp. 757–766.

[65] J. Véronis, "Hyperlex: Lexical cartography for information retrieval," *Computer Speech and Language*, vol. 18, no. 3, pp. 223–252, 2004.

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

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

[68] J. Gracia, R. Trillo, M. Espinoza, and E. Mena, "Querying the web: A multiontology disambiguation method," in *Proc. of International Conference on Web Engineering*, Palo Alto, California, USA, 2006, pp. 241–248.

[69] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[70] D. R. Radev and R. Mihalcea, "Networks and natural language processing," *AI magazine*, vol. 29, no. 3, p. 16, 2008.

[71] R. Mihalcea and D. Radev, *Graph-based natural language processing and information retrieval*. Cambridge University Press, 2011.

[72] N. Madnani and B. J. Dorr, "Generating phrasal and sentential paraphrases: A survey of data-driven methods," *Computational Linguistics*, vol. 36, no. 3, pp. 341–387, 2010.

[73] J. Bjerva, J. Bos, R. van der Goot, and M. Nissim, "The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity," *SemEval 2014*, p. 642, 2014.

[74] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer, 2006, pp. 177–190.

[75] M. R. Costa-Jussà and M. Farrús, "Statistical machine translation enhancements through linguistic levels: A survey," *Computing Surveys (CSUR)*, vol. 46, no. 3, p. 42, 2014.

[76] J. Berant, N. Alon, I. Dagan, and J. Goldberger, "Efficient global learning of entailment graphs," *Computational Linguistics*, 2015.

[77] S. Harabagiu and A. Hickl, "Methods for using textual entailment in open-domain question answering," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 905–912.

[78] E. Palogiannidi, E. Iosif, P. Koutsakis, and A. Potamianos, "Valence, arousal and dominance estimation for english, german, greek, portuguese and spanish lexica using semantic models," in *Proceedings of Interspeech, Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[79] N. Malandrakis and S. Narayanan, "Therapy language analysis using automatically generated psycholinguistic norms," in *companion submission to Interspeech*, 2015.

[80] E. Iosif and A. Potamianos, "A soft-clustering algorithm for automatic induction of semantic classes." in *INTERSPEECH*, 2007, pp. 1609–1612.

[81] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[82] F. Cazals and C. Karande, "A note on the problem of reporting maximal cliques," *Theoretical Computer Science*, vol. 407, no. 1, pp. 564–568, 2008.

[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] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 406–414.

[85] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *Proceedings of Human Language Technologies*. Association for Computational Linguistics, 2009, pp. 19–27.

[86] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, vol. 8, no. 10, pp. 627–633, 1965.

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

[88] Sofia, B.

[89] E. Bruni, N. Tran, and M. Baroni, "Multimodal distributional semantics." *Journal of Artificial Intelligence Research (JAIR)*, vol. 49, pp. 1–47, 2014.

[90] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: Computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th International Conference on World Wide Web*, 2011, pp. 337–346.

[91] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1406–1414.

[92] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[93] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

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

[95] E. Kreyszig, *Introductory functional analysis with applications*. Wiley. com, 2007.