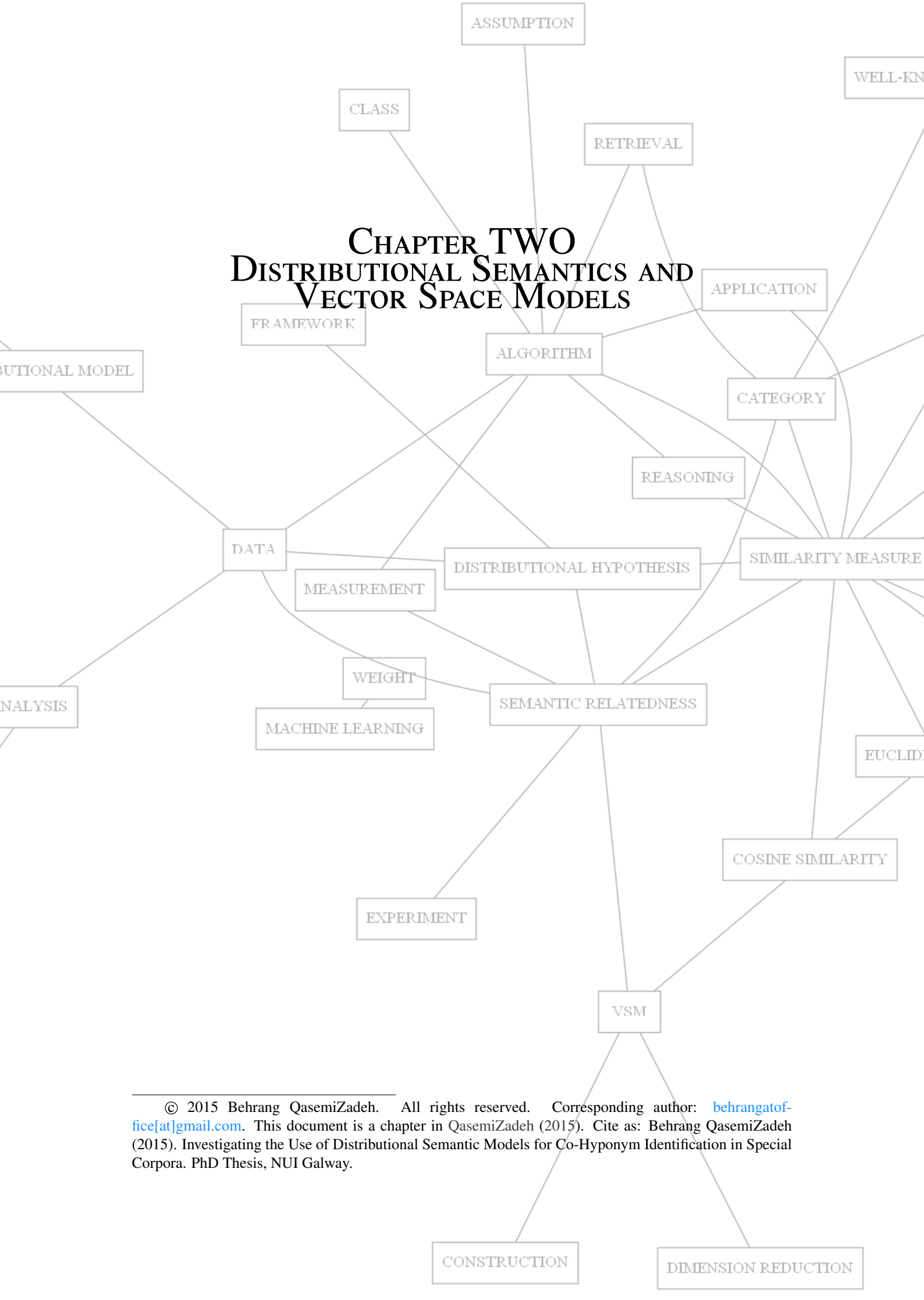



# Chapter TWO
# Distributional Semantics and Vector Space Models

 Corresponding author: behrangatoffice[at]gmail.com. This document is a chapter in QasemiZadeh (2015). Cite as: Behrang QasemiZadeh (2015). Investigating the Use of Distributional Semantic Models for Co-Hyponym Identification in Special Corpora. PhD Thesis, NUI Galway.

This page is intentionally left blank.

# Contents

This page is intentionally left blank.

# List of Figures

This page is intentionally left blank.

# List of Tables

This page is intentionally left blank.

# Chapter 2

# Distributional Semantics and Vector Space Models

Distributional approaches to semantics interpret the meanings of linguistic entities by investigating their distributional similarities in corpora. These empiricist corpus-based methods are often explained using Harris's (1954) *distributional hypothesis*. A vector space is an algebraic structure that can be employed to represent such distributional similarities. This representation of the distributional properties of linguistic entities generates mathematically well-defined models known as *vector space models of semantics*. In a vector space model, a distance formula measures semantic similarities between entities.

This chapter provides an overview of the distributional approaches to semantics. Section 2.1 provides a brief overview of distributional semantic models and the underlying distributional hypothesis. Section 2.2 introduces vector space models and provides mathematical preliminaries. The key processes for the discovery of meaning—that is, the steps from the construction of a vector space model to similarity measurements—are described in Section 2.3. In Section 2.4, the discussions are bound to the statistical learning theory. Finally, Section 2.5 concludes this chapter.

This page is intentionally left blank.

## 2.1 Distributional Semantics: Introduction

In order to provide a solution to the problems require a minimum level of text understanding, *distributional semantics* is a term that is often used to characterise a set of methods that rely on similarity-based reasoning frameworks. Distributional semantics embraces a number of approaches that employ similarity-based reasoning in an attempt to provide solutions to problems that require a minimum level of text understanding. Disregarding of the type of task and the way similarity-based reasoning is implemented, these methods aim to capture meanings of linguistic entities (e.g., words and phrases) from their usage in corpora. In distributional semantic models, therefore, meaning is a function of the distribution of linguistic entities in a given corpus.

Distributional semantics is motivated by the foundation of *structural linguistics* and the *distributional hypothesis*. The distributional hypothesis, which is often attributed to Harris (1954), presumes a correlation between distributional similarities of linguistic structures and their function in language (e.g., their syntactic role, meanings, and so on). Accordingly, distributional semantic methods suggest that the meanings of linguistic entities are established by the context in which these linguistic entities appear and their relationship to one another. For example, these methods suggest that the way words are distributed in text and co-occur with other linguistic expressions determines their meaning. Consequently, distributional semantics can be viewed as a statistical investigation of the co-occurrences of linguistic entities to capture their semantics from corpora and linguistic data.

Distributional semantics thus provides us with an *empiricist* and *quantitative* model of meaning in natural languages that is *context-dependent*. Compared to distributional semantics—on the other side of the spectrum of the methods that study semantics—*formal semantic* methods are motivated by a *rationalist* approach (e.g, see Partee, 2011). In these methods, the observation of language data is considered to be insufficient for gaining insight into the nature of language.[1] Hence, these methods rely on a priori knowledge that is often expressed in mathematical logic, for example, using the lambda calculus and predicate logic expressions (Blackburn and Bos, 2005). More importantly, compared to a distributional model that exploits an *inductive similarity-based reasoning*, formal semantic techniques rely on *deductive inference*.[2] Formal semantic models provide compelling tools and interesting model-theoretic methods to distil meaning from text. However, these methods can be used only after text is converted into logical expressions and a priori model of knowledge domain exists, which are a barrier to their use.

Table 2.1 lists several hypotheses that are embraced by the term distributional semantics. Despite the fact distributional semantics correlates differences in the meanings

[1] Put simply, rationalist approach sees the language as an *innate* object, an inherited capability (for a concise comparison see Manning and Schütze, 1999, chap. 1). In contemporary literature, these methods often attributed to Noam Chomsky, who collaborated with Harris as a doctoral student. Contemplating on this matter—although, out of the context of this thesis—will lead to questions such as *can we think without language?*, or *do we think independently of language?*

[2] As stated by Kamp (2002), although these methods are often studied by different communities, they can act as complementary tools for treating different aspects of the meanings in language and, thus, the problem of machine's understating of natural languages.

of linguistic entities to the differences in their distributional properties, it does not specify the variety of distributional information that should be taken into account. Moreover, the general idea of distributional semantics does not specify the type of meaning connotation that is attached to distributional differences. In order to establish a model that ties distributional similarity to meaning, therefore, two basic questions must be answered (see Sahlgren, 2006, chap. 3; Lenci, 2008; Baroni and Evert, 2009; Turney and Pantel, 2010):

- Which distributional properties of entities should be taken into account?
- How should different kinds of distributional properties be interpreted?

Different choices of distributional properties and their interpretation correspond to different kinds of models that capture different types of semantic similarities. Finding the appropriate answers for the above questions in a number of semantic computing tasks has formed a major empirical research theme known as distributional semantics.

### 2.1.1 Why Does Distributional Semantics Work?

In order to answer the question *why distributional semantics works*, I would like to begin with structuralism, an intellectual movement in the 1950s.[1] The essence of structuralism is to interpret human culture as a system of interconnected signs within a framework known as *semiotics* (see Chandler, 2007, for an introduction to the key concepts of semiotics). It was, perhaps, under the influence of the structuralism movement that Harris made his *distributional structure* proposal in order to justify the use of statistical techniques for natural language processing.[2] Particularly, Harris (1954) stated that

> the meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.

With a mathematical mindset, Harris elegantly restored the ideas dating back to linguists such as Ferdinand de Saussure (1857-1913). In this school of thought (i.e., structuralism), language is identified as an environment of interconnected elements and as a *functional system*. In simple terms, the elements of language are defined at different levels of abstraction and granularity and connected to each other through various relations. For instance, one may abstract language at morphological and phonemic levels, where words, morphemes, and phonemes can be considered as the building elements of language. The proposed relative perception in structuralism, then, allows elements of language to be identified by their relations to each other and not by their perceivable specification.

Structuralists apply the same fundamentals as stated above to lexical semantics. Lexical semantics is the study of the meaning of lexical units (see Paradis, 2012). According

[1] Readers, who wish to contemplate the (paradoxical) question asked here, are also invited to seek for answers in light of the *art of science* as explained by Dunbar (1996).

[2] For example, see reports from the *transformations and discourse analysis project* (http://www.cs.nyu.edu/cs/projects/lsp/pubs/tdap.html), which includes the development of the first English parsing program. See also Section 1.3 of the first chapter of this thesis.

| Reference | Articulation |
|---|---|
| Harris (1954) | difference of meaning correlates with difference of distribution |
| Firth (1957) | you shall know a word by the company it keeps |
| Rubenstein and Goodenough (1965) | words which are similar in meaning occur in similar contexts |
| Cruse (1986) | the semantic properties of a lexical item are fully reflected in appropriate aspects of the relations it contracts with actual and potential contexts |
| Miller and Charles (1991, cited in (Charles, 2000)) | the semantic similarity of two words is a critical function of their interchangeability, without a loss of plausibility |
| Morris and Hirst (1991) | word meanings do not exist in isolation. Each word must be interpreted in its context |
| Schütze and Pedersen (1995) | words with similar meanings will occur with similar neighbours if enough text material is available |
| Hanks (1996) | the semantics of a verb are determined by the totality of its complementation patterns |
| Lund and Burgess (1996) | word meanings as a function of keeping track of how words are used in context |
| Landauer and Dumais (1997) | a representation that captures much of how words are used in natural context will capture much of what we mean by meaning |
| Lin (1997) | the similarity between A and B, $sim(A, B)$, is a function of their commonality and differences |
| Lin and Pantel (2001) | if two (dependency) paths tend to occur in similar contexts, the meanings of the paths tend to be similar |
| Pantel (2005) | words that occur in the same contexts tend to have similar meanings |
| Sahlgren (2006) | words with similar distributional properties have similar semantic properties |
| Kilgarriff (2006) | word senses are abstractions from the data |
| Lenci (2008) | the degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear |
| Sinclair et al. (2004, cited in (Stubbs, 2009)) | there is a relation "between statistically defined units of lexis and postulated units of meaning" |

Table 2.1: Various articulations of the distributional hypothesis

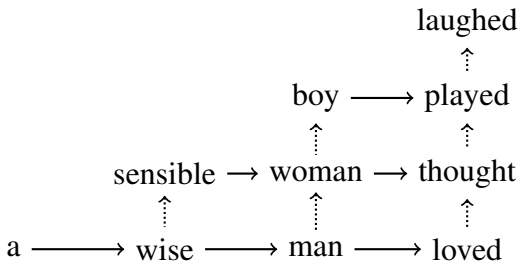



Figure 2.1: An illustration of syntagmatic and paradigmatic relations between words: the dotted lines show paradigmatic relations while solid lines represent syntagmatic relations.

to structuralists, the meanings of lexical units (e.g., words) are not substantial and self-subsisting, but a function of relations between them. Structuralists distinguish two types of relations between words: *syntagmatic* and *paradigmatic*. Furthermore, they assume that it is harmonious combinations of these paradigmatic and syntagmatic relations that convey meaning. Given this perspective, distributional semantic methods that model the meaning of lexical units identify significant patterns in this system of interconnected syntagmatic and paradigmatic relationships.

There is a syntagmatic relation between two words if they co-occur more frequently than expected by chance and if they have different grammatical roles in the sentences in which they occur. For instance, a semantic relation in the form of selectional restrictions between a verb and its arguments—such as the relation between *love* and *man* in the sentence *a wise man loved*—is an example of a syntagmatic relation. In contrast, the relationship between two words is paradigmatic if they can substitute one another in a sentence without affecting the grammatical acceptability of the sentence. For instance, for the given sentences *a wise man loved* and *a sensible woman thought*, the pair of words *man* and *woman*, *sensible* and *wise*, as well as *loved* and *thought* have a paradigmatic relationship. Paradigmatic relations may be contrastive associations, in which a group of words might constitute a paradigm. Synonymy and antonymy are examples of such paradigmatic relations. Figure 2.1 provides an illustration of syntagmatic and paradigmatic relations.

As stated by Lenci (2008) and Sahlgren (2008), a distributional semantic model that counts the co-occurrence of words captures a syntagmatic relationship between them. In this category of models, the co-occurring words in a window of text—such as a verse of a sentence, a sentence, a paragraph, etc.—define the context in which the relationship, thus the meaning, of words are induced. Models that extract multi-word expressions or those that specify syntactic or thematic relations between words are familiar examples in this category of distributional semantic models. In these models, the size of the region in which the co-occurrence frequencies are collected is an essential context parameter to be decided.

In contrast, if a distributional model counts the frequency of shared neighbours between words, then it captures a paradigmatic relation. In this category of models, words—or, in general linguistic entities—that appear surrounding a target word in text units such as

a window of text, sentence, and so on, define the context in which the meaning/relationship of these entities are induced. Models that detect synonymy relations or those that associate words to 'semantic categories'—for example, the proposed co-hyponymy identification task as well as the named entity recognition task that organises proper nouns into categories of persons, organisations, etc.—are familiar examples of these models. In this category of models, in addition to the size of text unit in which the co-occurrences are counted, the position of target entities (e.g., words) in relation to the context elements and the direction in which the neighbourhood extends are additional parameters that must be decided.

Let us now return to the question asked in the beginning: why do distributional semantic methods work? As described above, one of the major outcomes of conceptualising language as a functional system is that it can be studied empirically using *the scientific method*. As such, the question stated above is the point in which one of the limits of the scientific method is met. To understand this limitation, one must carefully distinguish between the three elements of fact (or, observation), hypothesis, and theory in the scientific method. Facts are inherently true;[1] in distributional approaches, they are equivalent to the *observations* made about linguistic phenomena that are modelled.[2] Since it is impossible to collect everything that language embraces,[3] conclusions are inevitably based on a number of selected observations. A *hypothesis* is an educated assumption. This assumption is made before designing experiments and collecting facts. If a hypothesis holds against a large number of observations, then the hypothesis is usually formulated as a *theory*. The induced theory is then employed to justify answers to a range of questions.

However, a theory can be rejected if new observations suggest this. Some relevant and unseen observations (or, their characteristics) that are important in the process of making a decision about the *truthfulness* of a hypothesis can be overlooked;[4] in turn, this can result in controversy.[5] Using the scientific method to model language and linguistic phenomena is certainly controversial. For the assessment of distributional hypotheses, given the complexity of natural language as well as its infinite and generative nature, simplifying characteristics of observations and experiments are inevitable. With this prelude, I suggest that, in fact, there is no definite answer to the question asked earlier: why do distributional semantic methods work?[6]

The first answer that seems plausible is that *distributional semantics works* because the underlying theoretical framework (i.e., usually the distributional hypothesis) is sound and effective. As stated above, the *success* of distributional semantics applied to a task depends on a number of parameters, most importantly on the appropriate identification

[1] Although very interesting, let us skip questions such as *what is truth?* in their philosophical sense (e.g., as discussed by Russell, 2014, chap. 3 and 4).

[2] Note that a number of prominent linguist object this statement.

[3] Since observations about most (if not all) linguistic phenomena are innumerable and hence it is impossible to record *everything* that is related to them.

[4] Or, observations can be *theory-laden*.

[5] There are well-known examples of this situation in the history of science, such as the *Mendel–Fisher* controversy (see Fisher, 1936) as well as the *Duhem–Quine* problem (see Stanford, 2013) to name a few.

[6] The short argument given here is discussed (*fairly*) by Eddington (2008) from a broader perspective that analyses the relationship between linguistics and the scientific method.

of linguistic elements and their relations within the problem context. As a result, if the *success* stories of distributional semantics are not sufficient to prove the effectiveness of the distributional hypothesis, they may also be insufficient for rejecting it.[1] Situating this discussion in the broader context that is given by Harris' sublanguages idea—as briefly mentioned in Chapter 1—can perhaps open new ways to discuss the question *why do distributional semantic methods work?*, asked here.[2]

By adopting an empiricist approach, the large number of experiments that confirm the ability of distributional methods (to address a range of tasks that require a level of language understanding) can be employed to verify the veracity of the distributional hypothesis.[3] Distributional methods have been successfully applied to information retrieval (e.g., Deerwester et al., 1990), semantic memory (e.g., Lund and Burgess, 1996), and word meaning disambiguation (e.g., Rapp, 2003), among others. These experiments have shown that contextual similarities can be employed to propose a reliable semantic model. However, distributional semantic models come with their own limitations and are still developing. The inability to handle traditional semantic notions such as *negation*, *scope*, *quantification*, and *compositionality* are examples of the distributional semantics limitations. Indeed, a number of these limitations arise from the constraints of similarity-based reasoning. Currently, these limitations are active research topics. Here, it is worth pointing out that the distributional hypothesis has not been employed to only justify distributional semantic methods. For example, a large amount of research in *speech recognition* and *language modelling* is based on the promise of the distributional hypothesis—that is, systemic functional perspective on language (even if it is not mentioned explicitly).

Distributional semantics is often praised for the practical method that it offers for constructing semantic models—that is, building frequency profiles from corpora. Developing a distributional model, therefore, requires minimal supervision; explicit human judgements are not usually required, and no rules need to be handcrafted. Consequently, compared to formal computational semantics, the development and maintenance of a distributional-based model are less time-consuming. More importantly, distributional semantic models equip us with two unique capabilities. As emphasised by Baroni (2013), distributional semantic models offer a systematic method to approximate degrees of similarity. In this framework, in contrast to formal models, semantic similarity is a quantitative prediction (e.g., a distance measure in a vector space). Such quantitative measures allow approximate degrees of similarity to be defined explicitly. This being the case, distributional models of semantics are capable of expressing semantic relatedness in a continuum of shades of *grey* instead of *black* or *white* (Baroni, 2013).

Secondly, distributional semantic methods permit meaning to be captured by arbitrary,

[1] Here, the notion of success is a source of controversy and ambiguity. While the discussion can be extended by describing the meaning of success, I assume success is defined by a tangible *figure of merit*—whether it is a simple quantitative measure used to evaluate an algorithm (e.g., *recall* in information retrieval tasks), or complex *Turing test-like* performance measures in more sophisticated tasks involved man-machine conversation. In fact, the definition of this performance measure (i.e., the definition of success in the given context) is an overlooked topic and can lead to flaws in the assessment of a hypothesis or unrealistic expectations or constitutions from observations in an experiment.

[2] Perhaps, by formulating and generalising the outcome of experiments more carefully.

[3] Yet, we do not like to curse one of a few tools that is available to us for analysing natural language.

heterogeneous, large-scale sets of symbols: from words in a lexicon to visual objects and scenes in images or a combination of these. For example, in order to improve a similarity measurement between words, Bruni et al. (2012) employ co-occurrence counts of words with a set of low-level image-based context elements. This is an exciting area of research considering the advances in wearable computing and the increasing availability of sensory information. As explained later, various techniques, such as random projections, enable distributional models to easily scale as demand requires. Compared to formal semantics, these properties make distributional semantic models a more desirable companion for the current paradigm shift in computing from algorithm-centric to data-driven approaches (e.g., see Zadeh, 2010).

### 2.1.2 Distributional Semantics and Principles of Interpretation

Distributional profiles and thus distributional semantics can be interpreted in, at least, two different representation frameworks: the probabilistic and vector space frameworks (Erk, 2012).[1] Distributional information consists of the counts of the co-occurrences of linguistic elements that can be stored and viewed in a tabular data format. This tabular data can be analysed either as a contingency table in a probabilistic modelling framework or in a vector space framework. These representation frameworks interpret and measure semantic similarity using different mechanisms.

A probabilistic-based model of distributional semantics employs probability theory and Bayesian mathematics. In this framework, a probabilistic inference indicates semantic similarity. A probabilistic approach associates linguistic entities with probability distributions based on the contexts that they appear in; it also calculates conditional and joint probabilities of contexts and elements. Eventually, a parameter estimation technique signifies semantic similarity. Latent Dirichlet Allocation (LDA) is a well-known example of a probabilistic approach to distributional semantics (Blei et al., 2003) .

On the other hand, vector space models construct a *metric* space from the given distributional profiles. Points in this metric space represent linguistic elements under consideration; a notion of distance between elements is defined and it indicates similarity between the elements. A 3-dimensional Euclidean space is probably the most intuitive understanding of such metric space. The vector space models thus results in a "geometrical metaphor" of meaning (Sahlgren, 2006). Landauer and Dumais's (1997) Latent Semantic Analysis (LSA) is a well-known example in this category of distributional semantic models.

Figure 2.2 summarises the discussion in this section. Although probability-based and vector space-based methods propose different conceptualisations of meaning (i.e., distributional probability vs. distance metrics), in essence, they are the same (e.g., see Turtle and Croft, 1992, in the information retrieval context). In both methods, meaning is derived from event frequencies presented by distributional profiles. However, throughout

[1] For instance, information-theoretic framework (e.g., as suggested by Resnik, 1995) and graph-based methodology (e.g., as employed in Navigli and Ponzetto, 2012) can be added to the list of representation frameworks for distributional semantic models.

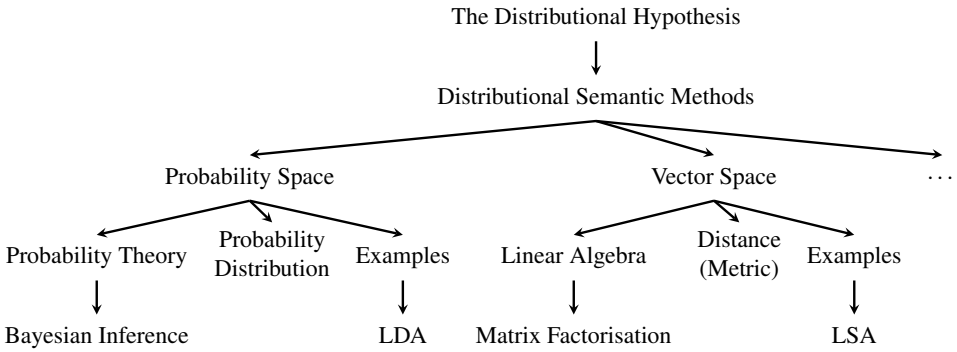



Figure 2.2: A mind map of different representation frameworks that can be employed for the implementation of a distributional semantic model.

this thesis, vector space models and distance metrics are employed to model semantic similarities. Following many researchers such as Widdows (2004) and Sahlgren (2006), it can be argued that the vector space models and the geometrical interpretation of the meaning are more intuitive than the probabilistic framework—for example, as put by Widdows (2004), *seeing is believing*. However, it is worth mentioning that these representation frameworks must be seen as complementary—such as the comparison of *generative* and *discriminative classifiers* (e.g., see the arguments in Nallapati, 2004, given in the context of information retrieval).

Last but not least, while distributional models of semantics can be presented using representation frameworks other than a vector space, a vector space can also represent semantic models other than distributional. For instance, Riordan and Jones (2011) use a *feature-based* model of semantics that is represented by a vector space. While distributional models are induced from statistical regularities of entities that appear in particular contexts (c.f., Section 2.2.2 for further details), feature-based models employ a rationalist approach and a set of descriptive features to reflect the meanings. As a result, although feature-based models of semantics can be presented by vector spaces, they are derived from an entirely different perspective on meaning. Therefore, not all the vector spaces necessarily implement distributional models of semantics.

The next section reviews basic mathematical definitions and notations that are used in vector space models.

## 2.2 Vector Space Models

### 2.2.1 Vector Space: Mathematical Preliminaries

In mathematics, an algebraic structure is a *set* together with one or more operations in it. Vector space is an algebraic structure that consists of a non-empty set and two binary operations that satisfy certain axioms. A vector space extends an algebraic structure called *field*. Informally, a field is a set of elements called scalars, or numbers, in addition to

two binary operations, and certain axioms that implement four familiar arithmetic operations of addition, multiplication, subtraction, and division over the set. The field of real numbers ($\mathbb{R}$) and the field of complex numbers ($\mathbb{C}$) are well-known examples.

A vector space can be denoted by a tuple

$$(V, F, +, \cdot). \tag{2.1}$$

The set $V$, whose members are called vectors, is defined over a field $F$ of scalars. For example, vectors can simply be a subset of a field such as complex numbers ($F = \mathbb{C}$, $V \subseteq \mathbb{C}$) or real numbers ($F = \mathbb{R}$, $V \subseteq \mathbb{R}$); or they can be an ordered sequence of scalars of a field such as $F = \mathbb{R}$, $V \subseteq \mathbb{R}^n$. The two binary operations are called vector addition ($V \times V \mapsto V : (\vec{v}, \vec{u}) \mapsto \vec{v}+\vec{u}$) and vector multiplication by scalars ($F \times V \mapsto V : (\alpha, \vec{v}) \mapsto \alpha \cdot \vec{v}$ ). The system $(V, F, +, \cdot)$ is a vector space if, and only if, it satisfies the following axioms:

- The binary operation addition $+$ forms an *Abelian group* over $V$. This implies the requirements of *Closure*, *Associativity*, and *Commutativity* for the binary operation $+$ over $V$, as well as the existence of *Identity* and *Inverse* elements in $V$.
- For the binary operation multiplication by scalars $\cdot$, $\forall \alpha \in F$ and $\vec{v} \in V$, $\alpha \cdot \vec{v} \in V$. In addition, if $\alpha, \beta \in F$ and $\vec{u}, \vec{v} \in V$, then $\alpha \cdot (\vec{u}+\vec{v}) = \alpha \cdot \vec{u} + \alpha \cdot \vec{v}$ and $(\alpha+\beta) \cdot \vec{v} = \alpha \cdot \vec{v} + \beta \cdot \vec{v}$.

Given a vector space $V$, if $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ are any vectors in $V$, and $\alpha_1, \alpha_2, \cdots, \alpha_n$ are any set of scalars in $F$, then

$$\alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \cdots + \alpha_n \vec{v}_n \tag{2.2}$$

is called a *linear combination* of the vectors. From the axioms, it can be shown that a linear combination of vectors in $V$ must belong to $V$. A set that contains all possible linear combinations of vectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ is called the *span* of $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$.

A set of vectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ from a vector space $V$ are called *linearly independent* if

$$\alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \cdots + \alpha_n \vec{v}_n = 0 \iff \forall i, \alpha_i = 0. \tag{2.3}$$

If $B = \{\vec{b}_1, \vec{b}_2, \cdots, \vec{b}_n\}$ is a set of linearly independent vectors in $V$, and $B$ spans $V$, then $B$ is called a basis of $V$. Consequently, vectors $\vec{v} \in V$ can be presented as a linear combination of the vectors $\vec{b}_i \in B$:

$$\vec{v} = \alpha_1 \vec{b}_1 + \alpha_2 \vec{b}_2 + \cdots + \alpha_n \vec{b}_n. \tag{2.4}$$

It can be proved that there exists at least one basis $B$ for $V$. The cardinality of $B$ is defined as the *dimension* of $V$. By limiting the focus to *finite-dimensional* vector spaces, the dimension of $V$ is thus the number of vectors in $B$[1]. The scalars $\alpha_1, \alpha_2, \cdots, \alpha_n$ in Equation 2.4 are called the *coordinates* of the vector $\vec{v}$ in that basis. It can be proved that the representation of a vector $\vec{v}$ in a basis $B$ is unique. The coordinates of elements of $V_n$ in a basis, subsequently, can be represented as a row or column matrix. Therefore, a collection of $m$ vectors in $V_n$ can be denoted by a matrix $\mathbf{M}_{m \times n}$, where the rows of $\mathbf{M}$ represent the vectors.

[1] From now on, an $n$-dimensional vector space is denoted by $V_n$.

In a vector space, additional structures are defined to quantify relationships between vectors. The fundamental concepts of *length* of a vector as well as *distance* and *angle* between vectors are the familiar geometrical interpretation of these structures.

A *norm* is a unary operation that associates a vector in $V$ with a scalar in $F$ (i.e., $V \mapsto F : (\vec{v}) \mapsto \|\vec{v}\|$) and satisfies the following axioms:

- Positivity, that is, $\forall \vec{v} \in V : \|\vec{v}\| \geq 0$;
- Definiteness, that is, $\|\vec{v}\| = 0 \iff \vec{v} = \mathbf{0}$;
- Homogeneity, that is, $\forall \vec{v} \in V$ and $\forall \alpha \in F : \|\alpha\vec{v}\| = |\alpha|\|\vec{v}\|$;
- Triangle inequality, that is, $\forall \vec{u}, \vec{v} \in V : \|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|$.

A vector space that is endowed with a norm is called a *normed vector space*. The norm of a vector $\vec{v} \in V$ (i.e., $\|\vec{v}\|$) is geometrically interpreted as the *length* of $\vec{v}$. The Euclidean norm—which is also called the $\ell_2$ norm—over the field of real numbers (i.e., $F = \mathbb{R}$) is the most familiar structure that satisfies the axioms listed above:

$$\|\vec{v}\|_2 = \sqrt{\Sigma_{i=1}^{n} v_i^2}. \tag{2.5}$$

Given the norm's definition, the distance $d(\vec{u}, \vec{v})$ between the two vectors $\vec{u}, \vec{v} \in V$ is given by

$$d(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|. \tag{2.6}$$

Given the Euclidean norm definition in Equation 2.5, respectively, the Euclidean distance—which is also called the $\ell_2$ distance—between the two vectors $\vec{v}$ and $\vec{u}$ in $V_n$ over$F = \mathbb{R}$ is given by

$$d_2(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|_2 = \sqrt{\Sigma_{i=1}^{n} (u_i - v_i)^2}. \tag{2.7}$$

In a similar fashion, an *inner product space* is a vector space that is equipped with an *inner product* structure. An inner product $\langle , \rangle$ is a binary operation that associates a pair of vectors in $V$ to a scalar in $F$ ($V \times V \mapsto F : (\vec{u}, \vec{v}) \mapsto \langle \vec{u}, \vec{v} \rangle$) and satisfies the following axioms:

- Positivity, that is, $\forall \vec{v} \in V : \langle \vec{u}, \vec{v} \rangle \geq 0$;
- Definiteness, that is, $\langle \vec{v}, \vec{v} \rangle = 0 \iff \vec{v} = \mathbf{0}$;
- Additivity for first element, that is, $\forall \vec{u}, \vec{v}, \vec{w} \in V$: $\langle \vec{u} + \vec{w}, \vec{v} \rangle = \langle \vec{u}, \vec{v} \rangle + \langle \vec{w}, \vec{v} \rangle$;
- Homogeneity for first element, that is, $\forall \vec{u}, \vec{v} \in V$ and $\forall \alpha \in F : \langle \alpha\vec{u}, \vec{v} \rangle = \alpha \langle \vec{u}, \vec{v} \rangle$;
- Conjugate interchange, that is, $\forall \vec{u}, \vec{v} \in V : \langle \vec{u}, \vec{v} \rangle = \overline{\langle \vec{v}, \vec{u} \rangle}$.

For $F = \mathbb{R}$ and the two vectors $\vec{u} = (u_1, u_2, \cdots, u_n)$ and $\vec{v} = (v_1, v_2, \cdots, v_n)$, a familiar structure that satisfied the above axioms is given using the standard multiplication of real numbers:

$$\langle \vec{u}, \vec{v} \rangle = \vec{u} \cdot \vec{v} = u_1 v_1 + \cdots + u_n v_n = \sum_{i=1}^{n} u_i v_i. \tag{2.8}$$

A geometric interpretation of the inner product and the norm gives the angle between the two vectors. In $F = \mathbb{R}$, the angle between the two vectors $\vec{u}$ and $\vec{v}$—that is $\theta$—is defined by the cosine inverse function:

$$\theta = \arccos\Big(\frac{\langle \vec{u}, \vec{v} \rangle}{\|\vec{u}\| \cdot \|\vec{v}\|}\Big). \tag{2.9}$$

It is proved that $-1 \leq \frac{\langle \vec{u}, \vec{v} \rangle}{\|\vec{u}\| \cdot \|\vec{v}\|} \leq 1$ and thus $\theta$ is always valid—that is, $\theta \in [0, \pi]$.

It is said that the two vectors $\vec{u}, \vec{v} \in V$ are *orthogonal* if $\langle \vec{u}, \vec{v} \rangle = 0$. A basis of $V_n$ is an *orthogonal basis* if the vectors in the basis are mutually orthogonal. Moreover, if the norm of all the vectors in an orthogonal basis is equal to the unit length, then the basis is called an *orthonormal basis*. An orthonormal basis of $V_n$ is called the *standard basis* (i.e., $S = \{\vec{s_1}, \cdots, \vec{s_n}\}$) of $V_n$ if each vector $s_i \in S$ has only one non-zero entry. It is common to represent $V_n$ by the coordinates of vectors in $S$, which is proven to be unique.

The given definition for vector space is inherently abstract and can be extended to a fairly arbitrary set of objects that forms a field. In addition, there are a number of definitions for the binary operations of addition, multiplication, and norm that satisfy the proposed axioms in vector spaces. Consequently, alternative structures for comparing vectors can be defined and used by changing the aforementioned components. In the context of distributional semantics, however, the employed vector space structures are usually limited to the *subspaces* of a *finite real space*, particularly, a *finite Euclidean space* $\mathbb{E}^n$.

A subset $W \subset V$ of a vector space is a *subspace* of $V$ if

- for each two vectors $\vec{w}_1$ and $\vec{w}_2$ in $W$, then $\vec{w}_1 + \vec{w}_2 \in W$;
- for any scalar $\alpha \in F$ and $\vec{w} \in W$, then $\alpha \cdot \vec{w} \in W$.

Given a finite positive integer $n$, the set of all ordered $n$-tuples $\vec{u} = (u_1, u_2, \ldots, u_n)$ of real numbers and the binary operations

$$(\vec{u} + \vec{v})_i := u_i + v_i \tag{2.10}$$

and

$$\alpha \cdot \vec{u} = (\alpha u_1, \alpha u_2, \ldots, \alpha u_n) \tag{2.11}$$

that are based on the real numbers' addition and multiplication form a *finite real vector space*, shown by $\mathbb{R}^n$. An $\mathbb{R}^n$ that is equipped with a Euclidean norm (see Equation 2.5), or by analogy with an inner product (Equation 2.8), is called a finite Euclidean space. As it will be discussed in Section 2.3.4, to compute similarities, $\mathbb{R}^n$ can be endowed with a norm structure other than the Euclidean norm.[1]

The vector space-based approaches to distributional semantics use the key concepts introduced in this section to model the meanings of linguistic entities. Given $n$ context elements, each element $\vec{s_i}$ of the standard basis of a vector space $V_n$ is employed to express an $i^{th}$ context element. Given $V_n$, in order to analyse the meaning of a linguistic entity, it is represented by a vector $\vec{v}$ as a linear combination of $\vec{s_i}$ and scalars $\alpha_i$, similar to what

[1] An elaboration of the discussed topics in this section can be found in William J. Gilbert (2004).

is shown in Equation 2.4. In this linear combination, the value of $\alpha_i$ is acquired from the frequency of the co-occurrences of the linguistic entity that $\vec{v}$ represents and the context element that $\vec{s_i}$ represents. As a result, the coordinates of $\vec{v}$ show the correlations between the linguistic entity that $\vec{v}$ represents and the employed context elements in the model (see Figure 2.3 as an example).

In this framework, a collection of $m$ linguistic entities whose meaning is being analysed using $n$ context elements builds a subspace of an $n$-dimensional vector space consisting of $m$ vectors. To compute similarities between the linguistic entities, this vector space is endowed by a structure such as inner product or norm. Subsequently, the angles or distances between vectors indicate the similarities of the linguistic entities that they represent. As stated earlier, often real numbers denote the magnitudes of the correlations between the linguistic entities and the context Respectively, the coordinates of vectors can be denoted by a matrix $\mathbf{M}_{m\times n}$ of real numbers. Each entry of $\mathbf{M}$, thus, represents the intensity of the relationship between a context element and an entity.

In order to distil the meanings of linguistic entities, a vector space will be the subject of several processes. Before introducing these processes in Section 2.3, the discussion continues with an elaboration of choosing the context elements in vector space models of distributional semantics.

### 2.2.2 Vector Space Models in Distributional Semantics

In natural language processing, vector space models (VSMs) are often identified by the model proposed in Salton et al. (1975). In the context of information retrieval (IR), Salton et al. employed a VSM to measure similarity between documents and queries. In the proposed model, natural language text documents, as well as natural language queries, are represented as vectors in a high-dimensional vector space. In this vector space, vectors that are close to each other are assumed to be semantically similar, while vectors that are far apart are semantically distant.

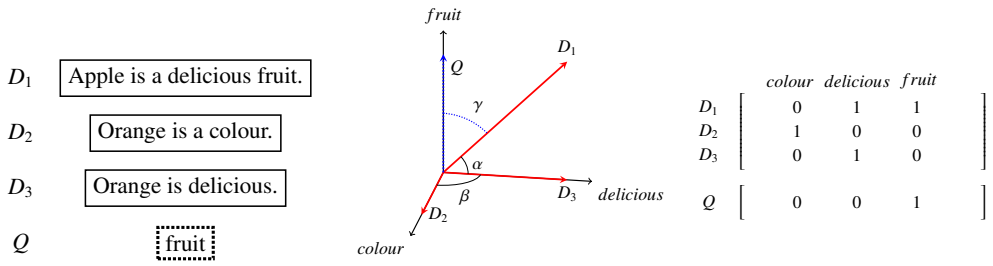
Given $n$ distinct terms $t$ and a number of documents $d$, in Salton et al.'s (1975) model, each document $d_i$ is represented by an $n$-dimensional real vector

$$\vec{d_i} = (w_{i1}, w_{i2}..., w_{in})$$

where $w_{ij}$ is a numeric value that associates the term $t_j$, for $1 < j < n$, to the document $d_i$. The numeric association between the term $t_j$ and the document $d_i$ may correspond to a *weighted* value, such as the frequency of terms in documents. Alternatively, it can be an un-weighted value restricted to 0 and 1. For a collection of $m$ documents, a *document-by-term* matrix $\mathbf{M}_{m\times n}$ denotes the constructed vector space.

A document-by-term VSM can be equipped by the inner product structure to quantify similarities between documents. Therefore, the similarity between the two documents that are represented by vectors $\mathbf{d}_i$ and $\mathbf{d}_j$ can be given by their *cosine similarity*:

$$sim(d_i, d_j) = \frac{\langle \vec{d_i}, \vec{d_j} \rangle}{||\vec{d_i}||||\vec{d_j}||} = \frac{\sum_{k=1}^{n} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2}}. \tag{2.12}$$




(a) Document Collection (b) VSM Illustration (c) Document by Term Matrix

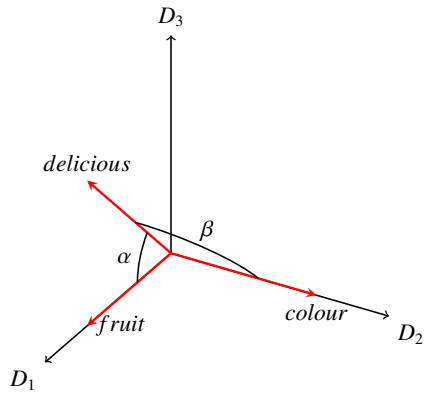
Figure 2.3: The VSM proposed by Salton et al. (1975): (b) shows a vector space that is constructed from the given document collection in (a). Words *fruit*, *delicious*, and *colour* are chosen as the context elements/terms and represented by the standard basis of the VSM. The vectors' elements denote the frequency of the terms in their corresponding documents. As is shown in (b), in this VSM, $D_3$ is more similar to $D_1$ than $D_2$ ($\alpha < \beta$). The given input query $Q = fruit$ is also represented by a vector. $Q$ is closer to $D_1$ than to other documents ($\gamma < \frac{\pi}{2}$). Figure (c) shows the *document by term* matrix denotation of the constructed VSM.

In the above equation, similar to Equation 2.9 in Section 2.2.1, the numerator is the dot product of the vectors and the denominator is the multiplication of the Euclidean length of vectors. The genius of the Salton et al. method is that queries, in a retrieval task, are treated as pseudo-documents and are represented by vectors too. In a vector space constructed from a document collection $C$, the most similar documents to a query $q$ (such as a *keyword*) are found by computing $sim(q, d)$ for all the documents $d \in C$ (Figure 2.3).

The VSM described above implements a hypothesis known as the *bag of words*. The BoW hypothesis suggests that the relevance of documents can be assessed by counting words that appear in the documents, independent of their order or syntactic usage patterns. Documents with similar vectors in a document-by-term model, therefore, are assumed to have the same meaning. However, in order to implement a distributional hypothesis other than BoW, a VSM can be generalised to sets of entities other than documents and sets of context elements other than words that appear in documents.

Deerwester et al. (1990) showed that similarity between words can be captured by transposing the *document-by-term* matrix into a *term-by-document* matrix.[1] The proposed model by Deerwester et al. (1990), called latent semantic analysis (LSA), hypothesises that terms that are semantically similar occur in collections of similar documents. In this term-by-document model, the cosine similarity of vectors, which represent terms, can be employed to indicate the semantic relatedness between terms. The same model as the LSA was introduced much earlier by Jones (1972) (cited in Wilks and Tait, 2005); the novelty of the LSA, however, is the use of *singular value decomposition* (i.e., a matrix factorisation technique) for the arrangement of context elements at a reduced dimensionality (see Section 2.3.3). As described later in Section 2.3.3, singular value decomposition is a matrix factorisation technique, which allows irrelevant context elements to be eliminated from a vector space in order to enhance the similarity measures.

[1] From now on, the terms vector space, context vectors, and context matrix may be used interchangeably.




(a) Illustration of the Vector Space

| | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|
| *delicious* | 0 | 1 | 1 |
| *fruit* | 1 | 0 | 0 |
| *colour* | 0 | 1 | 0 |

(b) Term-by-Document Matrix

Figure 2.4: A vector space model that is constructed from the document collection given in Figure 2.3. The three documents $D_1$, $D_2$, and $D_3$ are the context elements. Therefore, the basis of the vector space represents each of them. The vectors represent words/terms, in which the coordinates of the vectors indicate the co-occurrence relationships between the words/term and documents. In the given example, cosine similarities between the vectors suggest that *delicious* is semantically more related to *fruit* than to *colour* (i.e., $\alpha < \beta$ in Figure 2.4a). Figure 2.4b shows a matrix denotation of the constructed *term-by-document* model.

The *term-by-document* model can be further generalised by replacing documents with text of an arbitrary length, such as a word, or window of words of a certain size. For instance, the proposed method in Lund and Burgess (1996) captures the semantic similarity of words using a *word-by-word* vector space. The resulting word-by-word model takes the co-occurrences of words as a measure of similarity. Even lexico-syntactic patterns can be employed to define context elements. VSMs, thus, can be categorised and studied according to the type of context element that they employ and the linguistic entities that they represent (e.g., as suggested by Turney and Pantel, 2010; Baroni et al., 2010). As discussed, the type of context elements and the linguistic entities in a model is determined by the model's underlying hypothesis and intended application.

### 2.2.3 Types of Models and Employed Context Elements

Distributional semantic models and the employed context elements for their construction can be categorised and studied from several overlapping perspectives.

First, these models can be categorised by the type of semantic relationship that they target—that is, whether they characterise syntagmatic or paradigmatic relations between the linguistic entities in the model (see also Sahlgren, 2006, chap. 7). As discussed earlier, in Section 2.1.1, the context elements, thus dimensions of a vector space model that captures a syntagmatic relation between linguistic entities, show the magnitude of the frequency of the linguistic entities that co-occur in text. For instance, models that are used to measure lexical semantic relatedness (e.g., as employed in Jurgens et al., 2012) must capture a syntagmatic relation. However, in a model that captures a paradigmatic relation

between linguistic entities (e.g., a model that discovers the synonym or the hypernym relationship), the context elements show the neighbourhoods that are shared between the linguistic entities.

As implied in Baroni et al. (2010), distributional semantic models can be also categorised according to the approach that they employ to distil co-occurrence frequencies. A distributional method results in a so-called *flat* or *unstructured* model if the process of collecting co-occurrence frequencies in text is coincident with neglecting linguistic information such as part-of-speech tags or syntactic relations.

To implement a flat model that collects the co-occurrence frequencies of linguistic entities—that is, to capture a syntagmatic relationship—the only parameter that needs to be verified is the size of the text region in which the co-occurrence is regarded. Deerwester et al.'s (1990) LSA is an example of a flat model that captures syntagmatic relations between linguistic entities. In LSA, the text region is of the size of logical documents. Lund and Burgess (1996) present another example of a flat model that captures a syntagmatic relation between words, however, it uses a narrow text region (i.e., a text window of $n$ words for $n = 10$ in the reported experiment). As a rule of thumb, Sahlgren (2006, chap. 9) suggests that a wide text region tends to show a better performance than narrow text region if syntagmatic relations are approximated; inversely, the use of narrow text regions for collecting co-occurrences of the neighbourhoods that are shared between linguistic entities has a better performance than using wide text regions when paradigmatic relations are approximated.

When a flat model collects the co-occurrence frequencies of the neighbourhoods that are shared between linguistic entities (i.e., to capture a paradigmatic relationship), however, the *direction* in which text region are extended is also important. Text regions can be stretched (a) only to the left side of a linguistic entity to collect the co-occurrences of the linguistic entity with its preceding words, (b) only to the right side to collect co-occurrences with the succeeding words or (c) around the linguistic entity (i.e., in both left and right directions). If text regions are extended around linguistic entities, then the position of the linguistic entities in the text region (*symmetry*) is an additional parameters that can be changed.

The *order* of words in the text regions can be also important. To capture the word order information in a model, the appearance of distinct words in distinct positions in text regions must be distinguished—for example, by appending additional dimensions to the model. The words' order information may be also encapsulated implicitly using $n$-gram sequences, or using an additional vector structure—for instance, as suggested in Jones and Mewhort (2007). Section 5.3.2.3 of Chapter 5 will describe the *permutation technique* and justify it mathematically, which will be employed later in this thesis. This method is first suggested by Sahlgren et al. (2008) for the incorporation of word order information in the vector space models that are built using the random indexing technique.

Curran (2004, chap. 3) distinguishes flat models by the way that they treat logical text boundaries such as sentence and paragraph boundaries. The width of text regions may be fixed irrespective of logical text segment boundaries, or it may be restricted by them. In the first case, text regions can be expanded to two or more logical text segments. Last but not least, words in flat contexts can be presented in their *stemmed/lemmatised* form

to build *stemmed* models (as named by Murphy et al., 2012). The reported experimental results are contradicting with respect to the significance of the inclusion of word order information as well as lemmatisation in the performance of distributional models (e.g., see Bullinaria and Levy, 2012).

*Linguistically aware* models, which are also called *structured models*, are the second category of the models that are proposed in Baroni et al. (2010). In this models, text regions are first annotated with linguistic information such as part-of-speech tags or syntactic relations. These linguistic annotations may be captured by the model, or it may be used to filter a number of co-occurrences. Linguistically aware models are used based on the intuition that linguistic information provides a stronger cue of semantic similarity than flat models. For instance, a window of words with particular part-of-speech categories, namely nouns, adjectives, and verbs, form the context proposed in Baroni et al. (2010). Widdows (2003) and Jonnalagadda et al. (2012) are other examples that employ part-of-speech tags in order to filter co-occurrences.

Pioneered by Grefenstette (1994), a sub-category of linguistically aware models is defined by the use of syntactic relations. In its simplest form, pairs of dependency relations $Dep_r$ and words in text regions $C_w$ (i.e., $(Dep_r, C_w)$) form syntactic contexts. In this model, the co-occurrence frequencies are induced by observing target words/entities that are in particular $Dep_r$ relationships with $C_w$. Syntactic contexts, however, may correspond to more complex syntactic *paths* than that described here. Padó and Lapata (2007) argue that syntactic structure in general and argument structure in particular are close reflections of the lexical meanings. Several experiments suggest that syntactic-based models can outperform flat models (e.g., see Erk and Padó, 2008; Jurgens and Stevens, 2010; Thater et al., 2010; Séaghdha and Korhonen, 2011; Weeds et al., 2014).

The third group of models, which can be called *attribute-value*-based models, are those that collects the co-occurrences of linguistic entities and particular *lexico-syntactic patterns*. As mentioned by Baroni et al. (2010), lexico-syntactic patterns are often handcrafted and used to capture concept associations, in particular semantic analysis task such as detecting an entailment relation. For instance, a context may be defined as the presence of the lexical pattern "X *such as* Y" between the two entities $X$ and $Y$ in order to indicate a subordinate relation between them. The main assumption here is that a surface pattern can be an indication of the presence of semantic relations. An example of this type of model is suggested by Hartung and Frank (2010).

The types of models that are listed above can be populated by the text kernel methods that are often used in text classification task. A well-known example is a string kernel (Lodhi et al., 2002). Models that are built using text kernels can be placed in one of the categories listed above, depending on the type of the employed kernel. For example, the resulting model from the application of a string kernel is often a flat model. Using a tree kernel such as the one proposed in Collins and Duffy (2002), however, results in a structured model. Other types of kernels in applications other than text classification are also conceivable (e.g., see Plank and Moschitti, 2013; Mehdad et al., 2010).

The methods that are employed for collecting co-occurrences are not restricted to the above-listed categories. A number of recently employed methods for the construction of distributional semantic models can be categorised as those that use *extra-linguistic* context

elements. As explained earlier in Section 2.1.1, the notion of the context element can be extended to sets of objects other than text. For example, in Bruni et al. (2012), low-level visual features enrich a VSM that measures semantic similarities between words (see also Bruni et al., 2014). Similar *extra-linguistic*-based models are employed in Chen et al. (2012); Roller and Schulte im Walde (2013); Silberer et al. (2013). As suggested in Anderson et al. (2012), recent research results (e.g., Mostow et al., 2011; Mitchell et al., 2008) further validate the suitability of *extra-linguistic*-based models for semantic modelling from the cognitive point of view.

Other trending usage examples of *extra-linguistic* context elements, although less exciting than the above list, are found in the context of the Web. Openly available knowledge bases on the Web are rich sources of *extra-linguistic* information and have served an increasing number of distributional models. For instance, the explicit semantic analysis (ESA) technique builds a *term-by-document* model with *extra-linguistic* context elements that are derived from the topical structure of a knowledge base such as Wikipedia (Gabrilovich and Markovitch, 2007). Reversely, Angeli and Manning (2014) employ a distributional model and the structured data in open-domain knowledge bases to enable common sense reasoning, however, for new and unseen entities. In a similar line of research, Gardner et al. (2014) use similarities in a vector space model to enhance reasoning over knowledge-bases.

The list presented here is endless. Table 2.2 lists a number of distributional models and their applications. The type of model and the employed method for collecting co-occurrences is determined by the underlying hypothesis and the task in hand. A new task implies a new hypothesis, and a new hypothesis often demands a new method for collecting co-occurrences and thus a new type of model. In short, the construction of flat models is computationally less expensive. However, flat models are often high-dimensional, which in return may result in a high computational cost for similarity measurement. Such VSMs may include uninformative, and sometimes irrelevant, context elements, which can reduce the performance of the model. The use of linguistic information may prevent the problems mentioned above, however, at the expense of higher computational costs for VSM construction. However, the use of linguistic information may introduce a level of noise that is originated from the use of linguistic analysis tools. If the co-occurrences are filtered by linguistic information or lexico-syntactic patterns, then a larger amounts of text data might be required to avoid the *sparsity* in the constructed models. Depending on the anticipated application for the constructed model, the use of a structured model may not necessarily enhance the results (e.g., as reported in Zeng et al., 2014).

## 2.3 Processes in Vector Space Models

The construction of vector space models of semantics and the task of meaning discovery involve a set of processes. These processes vary from one application and model type to another. However, a general pattern of processes can be identified in most of the applications of VSMs: a three-step pre-process followed by a four-step process (Turney and Pantel, 2010).

| Reference | Model/Type/Application Domain |
|---|---|
| Salton et al. (1975) | document-by-term model, flat<br>in *information retrieval* |
| Deerwester et al. (1990) | term-by-document model, flat<br>in *information retrieval* |
| Lund and Burgess (1996) | word-by-word model, flat<br>a text window of 2 words to the left and right of each target word<br>as *a representational model of semantic memory* |
| Lin (1998a) | word-by-word model, linguistically aware<br>words in syntactic relations with target words<br>in *thesaurus construction*, *automatic detection of similar words* |
| Lin and Pantel (2001) | "path"-by-word, linguistically aware<br>words in syntactic relations with automatically induced lexico-syntactic patterns (path)<br>entitites are constrained paths in dependency tree<br>in *unsupervised inference rules discovery* |
| Kanejiya et al. (2003) | word-by-word model, linguistically aware<br>part of speech (PoS) tagged words, blocks of POS tag information around a target word<br>in *automated essay scoring* |
| Widdows (2003) | word-by-word model, lingustically-aware<br>words surrounding a target word<br>target words discriminated by PoS tags<br>in *taxonomy extraction* |
| Padó and Lapata (2007) | word-by-word model, linguistically aware<br>pair of words and dependency relations (anchored paths)<br>in *synonym detection*, *semantic priming*, and *sense disambiguation* |
| Gabrilovich and Markovitch (2007) | term-by-document model, extra-linguistic-based<br><br>concepts that are derived from the Wikipedia' articles<br>in *information retrieval*, *document similarity*, and *word relatedness* |
| Baroni et al. (2010) | concept-by-attribute-value model, attribute-value-based model<br>lexico-syntactic patterns using PoS tags and dependency structures<br>in *concept description extraction* |
| Jonnalagadda et al. (2010) | word-by-word model, lingusitcally-aware<br>symmetric text window, PoS tags, encoded words' order<br>in *named entity recognition* |
| Séaghdha and Korhonen (2011) | word-by-word model, linguistically aware<br><br>context elements derived from dependency structure<br>in *lexical substitution ranking* |
| Hartung and Frank (2011) | word-by-attribute model, attribute-value-based<br>adjectives and nouns with context elements that are induced using an LDA topic model algorithm<br>in *attribute selection for Adjective-Noun* |
| Lops et al. (2013) | term-by-meta-document model, extra-linguistic-based<br>textual metadata derived from web resources, URLs, HTML meta-tags, social bookmarks<br>in *tag recommender systems* |
| Anderson et al. (2013) | word-by-bag-of-visual-words model, extra-linguistic-based<br>textual models, verbs and textual windows of fixed size, augmented with image-based features,<br>to *study the correlation between fMRI-based neural patterns and distributional semantic measures* |

Table 2.2: Examples of the employed context elements in vector space models of semantics in different application domains

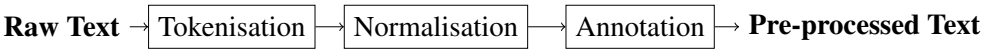



Figure 2.5: Pre-processes to vector space construction

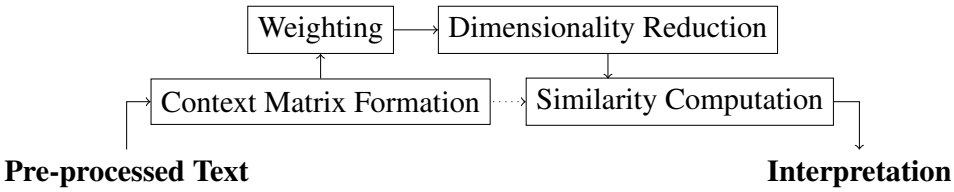



Figure 2.6: From frequency to meaning: a common four-step process flow in vector space models

As shown in Figure 2.5, pre-processing starts with a text *segmentation* and *tokenisation* process in order to detect linguistically well-defined text boundaries such as words and sentences from an input text collection (see Palmer, 2010). The successive normalisation process may organise similar entities or filter some of them. For example, a simple normalisation process may convert all characters to lowercase, convert words to their lemmatised form, or remove some of the tokens such as stop words. Finally, an annotation process augments text units with additional information. For example, PoS tagging and syntactic parsing are common annotation processes.

Pre-processed data usually undergoes a four-step process that start with the collection of co-occurrences and the calculation of event frequencies and ends with an interpretation of the calculated similarity measures (Figure 2.6). In the first step, the frequency of the co-occurrences of linguistic entities and context elements is calculated, and vectors that represent linguistic entities are built. Non-compulsory processes of *weighting* and *dimensionality reduction* may follow the construction of context vectors. The process is finished by a method that measures similarity between the constructed vectors. Although these steps are listed back-to-back, in practice, they may be combined or skipped, as discussed in the following sections.

### 2.3.1 Context Matrix Formation: Collecting Co-Occurrences

Context matrix formation determines numeric associations between linguistic entities and context elements. In its simplest form, this association is an un-weighted binary value restricted to 0 and 1,[1] and it shows the absence or presence of the occurrences of a linguistic entity with a context element. In a typical term-by-document model, for instance, un-weighted associations indicate the presence of a term (linguistic entity) in a document (context element) using value 1. However, the association between linguistic entities and context elements can be a weighted value. The weighted associations usually correspond to the frequency of the observation of the co-occurrences of linguistic entities and context elements. For example, in a term-by-document model, the frequency of the occurrences of terms in documents can specify a weighted value.

[1] That is, $F = \{0, 1\}$, in the given Tuple 2.1.

Context matrix is often instated using a sequential scan of input text-data, for example, by collecting the co-occurrence frequencies in a hash table or database. Alternatively, a search engine that keeps an inverted index of context elements and linguistic entities can be used (Turney and Pantel, 2010). The collected frequencies in tabular presentations are then converted to an efficient data structure—for example, a dictionary of keys, list of lists, and so on—that are often used for sparse matrix representation and manipulation (for an introduction to such data structures see Barrett et al., 1993, chap. 4). However, further complications may be imposed by the adapted approach for collecting co-occurrence frequencies. For instance, Schütze (1998) employs a method called *context-group discrimination* that goes beyond counting the co-occurrence frequencies and building context vectors at once.

An alternative set of vector space construction methods may not directly count the co-occurrence events and build a co-occurrence frequency matrix. For example, Gallant (2000) suggests a three-stage process for the construction of a vector space model. In the first step, each word, which is assumed to be an irreducible context element that captures meaning, is assigned to a normalised random vector. In the second step, using an iterative process similar to the training in Kohonen's self-organising maps, vectors of adjacent words are altered in an attempt to preserve and show the neighbourhood relationships. Finally, the vector space is generated using a combination of these vectors such as their weighted sum (see also Gallant, 1991, 1994, for more details).

Kanerva et al. (2000) propose a similar method for vector space construction, which is called *Random Indexing* (RI). The RI technique constructs a vector space using a similar a two-step process and in a fashion to Gallant's (2000) method. In the RI technique, the process of vector space construction is carried out by the accumulation of a set of *randomly*[1] generated sparse vectors, called *index vectors*. Each index vector represents a context element in the model. To collect the co-occurrences, a linguistic entity is first assigned to an empty vector that has the same dimension of index vectors. The co-occurrence of a linguistic entity and a context element is then captured by accumulating the index vector that represents the context element to the vector that represents the linguistic entity. A similar technique, named *TopSig*, is proposed by Geva and De Vries (2011). In these methods, context matrix formation merges with the dimensionality reduction step, often to address scalability issues that are associated with processing large corpora. These methods are studied in depth in Chapter 4.

### 2.3.2 Weighting

The construction of a context matrix is usually accompanied by a weighting process in order to minimise the effect of the bias that may result from simple co-occurrence counting. The major sources of bias are frequent context elements and entities. Frequent context elements that are associated with greater numeric values can dominate those context elements with smaller numeric values. In a similar way, more frequent linguistic entities may be associated with a larger number of context elements. Both of the above scenarios cause bias. The amount and effect of this bias is dependent on the employed method for

[1] See Chapter 4 for an explanation of the meaning of *random* in this context.

the similarity measurement.

The above reasons for weighting can be viewed, by some analogy, in conjunction with *feature selection* in machine learning community (e.g., see Turney and Pantel, 2010, take on the topic).[1] First, it is desirable to give higher weights to more discriminative but less frequent context elements. For example, in an information retrieval (IR) framework that employs a document-by-term model, using the *raw term frequencies* (tf) implies the same significance of terms when measuring the similarity between documents. However, the *term frequency–inverse document frequency* (tf-idf) measure can substitute the raw term frequencies in order to give higher value to more discriminative terms. The tf-idf measure normalises raw term frequency weights by the inverse document frequency of terms (idf): tf-idf = tf $\times$ idf. The idf of a rare term, which assumes to be a discriminative context, is high, while the idf of a frequent term is expected low (for more details on tf-idf weighting in IR context, see Manning et al., 2009, chap. 6).

For types of models other than document-by-term, tf-idf can be replaced by a *measure of association* that indicates the strength of relationships between entities and contexts. As verified in Curran (2004, chap. 4), context elements with stronger correlations to linguistic entities are more informative than contexts with weaker correlations. A weak association between a context element and a linguistic entity implies their independence from each other. However, a strong association suggests that changes in a context element are likely to occur with changes in linguistic entities, thus, the context element discriminates between the linguistic entities well. For instance, in a term-by-term model, the *point-wise mutual information* measure can be replaced by the simple term co-occurrence counts (see, e.g., Bullinaria and Levy, 2007, for further explanation and experimental comparison). Subsequently, the calculated associations can be used to sort the context elements by their importance, and if desirable to filter a number of them.

Second, the weighting process is leveraged by a method often called *length normalisation* to cancel bias that results from highly frequent linguistic entities. For example, in an IR document-by-term model, length normalisation corresponds to techniques that cancel the advantage of long over short documents in retrieval tasks. Long documents tend to appear with many terms; additionally, long documents are likely to have large term frequencies (Singhal et al., 1996). In this setting, length normalisation adjusts the term weights in conformity with the length of documents. The length normalisation, however, can be widened to any set of linguistic entities. In this generalisation, the frequency of entities is replaced by the exemplified document length. In line with this reasoning, highly frequent linguistic entities are likely to appear with more context elements than less frequent ones. Moreover, the context elements that are occur with highly frequent entities are probably associated with greater weights.

Among techniques that can be used for length normalisation, *unit-length normalisation* is a common approach. First, the length of a vector—that is, its norm—is computed. Then the collected frequencies for context elements in the vector are divided by the its computed. For instance, in an $\ell_2$-normed space, the length of vector $\vec{v}$, which represents a linguistic entity, is given by $\|\vec{v}\|_2 = \sum_i^{|\vec{v}|} \sqrt{v_i^2}$. To perform the unit length normalisation,

[1] In this context, the weighting process is often called *feature scaling*.

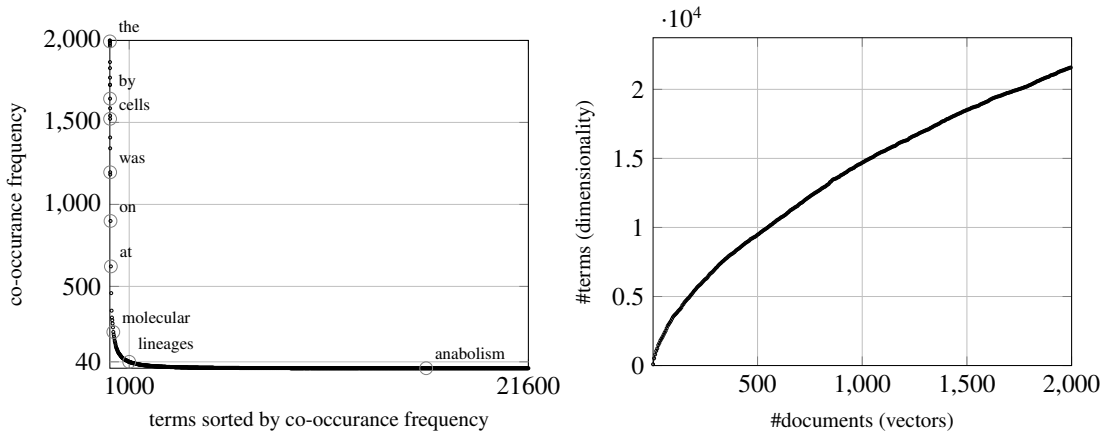
each element $v_i$ of $\vec{v}$ (which represents a context element) is divided by $\|\vec{v}\|_2$. Thus, the element $v_i{}'$ of the new normalised vector $\vec{v'}$ is given by vector $\vec{v'}$ (i.e., $v_i{}' = \frac{v_i}{\|\vec{v}\|_2}$). The impact of unit length normalisation varies from one task to another, and it depends on a number of additional factors, namely, the size of corpus and the distribution of entities and context elements such as suggested by Périnet and Hamon (2014b); Gorman and Curran (2006), and the employed metric for similarity measurement (see also Clark, 2015). These two factors are inspected later.

Contrariwise, weighting may be used to introduce intentional bias toward the co-occurrences of linguistic entities and certain context elements. For example, in a term-by-term model that counts the co-occurrences of words, Lund and Burgess (1996) assume that context words in closer vicinity to a target word represent more of its semantics than distant words. Therefore, the co-occurrence of words are weighted according to their distance in an inverse relation. For a context window of $n$ words on each side of the target words, the number of intervening words between the target and context words is defined as their distance $d$, and the frequency of occurrences are weighted with respect to their position in context windows by the magnitude of $n - d$ (Burgess, 2001). By the same token, Sahlgren et al. (2003) employ the function $2^{1-d}$ for the weighing of a context window.

Baroni et al. (2007) employ a weighting procedure to encode distributional histories of context words in a term-by-term model. The vectors are weighted using a ratio of the encountered frequencies of context words. Baroni et al. (2007) suggest that frequent words tend to co-occur with other words by chance. As a result, more frequent context words have less informative distributional history than rare context words. The employed weighting function, therefore, defines the influence of context words in an inverse proportion to their frequencies. Mathematically speaking, this method implements a Laplace smoothing of the collected co-occurrences, which can be also found in Turney and Littman (2003).

Zhitomirsky-Geffet and Dagan (2009) suggest that semantically similar words are best described by the contexts that are common between them. Therefore, they employ weighting to promote such contexts using a three-step *bootstrapping* process, similar to the proposed method in Bins and Draper (2001). At first, similarity values between words are calculated using contexts that are weighted by a mutual information measure. Next, the common contexts between the obtained set of similar words are promoted by increasing their weights. Yamamoto and Asakura (2010) propose a techniques that is bases on a similar idea. Finally, the similarities are recomputed using the updated weights. These methods can be criticised for their computational complexity, which is imposed by repetitive calculation of similarity measures, and then finding and sorting the common context elements. This procedure of weighting is also the fundamental idea behind the *learning process* in methods that employ neural networks such as Mikolov et al. (2013); Zeng et al. (2014); Irsoy and Cardie (2014).[1]

[1] Although, advances in technology, such as the availability of graphics processing unit accelerated technology, may remove this critique.




(a) The long-tailed distribution of words (b) Changes in the dimensionality

Figure 2.7: Zipfian distribution of the co-occurrences of linguistic entities and context elements: the distribution of word occurrences in documents a document-by-word model constructed using the GENIA corpus. In (a), the vocabulary is ranked by the frequency of the words' occurrences in the documents. As is shown, most of the words are rare, which results in a long-tailed distribution. Figure (b) shows the increase in the dimensionality of the model when new documents are.

### 2.3.3 Dimensionality Reduction

As discussed earlier, in distributional semantics, the distributional properties of linguistic entities—that is, their co-occurrences with various context elements—are compared to quantify some sort of semantic similarities. When a vector space is used to represent and analyse these distributional properties, each element of the standard basis of the vector space—that is, informally, each dimension of the vector space—represents a context element. Consequently, given $n$ context elements in a model, each linguistic entity in the model is expressed by an $n$-dimensional vector.

As the number of linguistic entities that are being modelled in the vector space increases, the number of context elements that are required to be utilised to capture and represent their meaning escalates (see the example in Figure 2.7). However, the proportional impact of context elements on semantic similarities lessens when their number increases. In a high-dimensional model, unless most coordinates of vectors are significantly different, it becomes difficult to distinguish semantic similarities. For instance, under certain broad conditions, it is likely that most entities are located at almost equal distances from each other (Beyer et al., 1999). Consequently, the proximity of linguistic entities may not express their semantic similarities.

For instance, in a word-by-document model that consists of a large number of documents, a word appears only in a few documents, and the rest of the documents are irrelevant to the meaning of the word. Few common documents between words results in sparsity of the vectors; and the presence of irrelevant documents introduces noise. These setbacks, which are caused by the high dimensionality of the vectors, are colloquially

known as *the curse of dimensionality*.

This curse of dimensionality is often explained using power-law distributions of linguistic entities and context elements—for example, the familiar Zipfian distribution of words (see Yang, 2013, for further description of power-law distributions). Zipf's law states that most words are rare while few words are used frequently. As a result, irrespective of the input data size, extremely high-dimensional vectors, which are also *sparse*—that is, most of the elements of the vectors are zero—represent linguistic entities.[1] For example, Sahlgren (2005) suggests that 99% of the elements of a vector in a typical word-by-word model are zero (see also Sahlgren, 2006, chap. 4).

A *dimensionality reduction* process lessens noise and improves the performance of the similarity measurement by reducing the number of context elements employed for the construction of a vector space. Dimensionality reduction can be performed by choosing a subset of context elements and eliminating the rest using a *selection process*. To resolve the curse of dimensionality and reduce the sparsity of a vector space, a selection process chooses a number of context elements that account for the most discriminative information in the vector space. Consequently, the selection process results in a vector space of lower dimension constructed by a subset of the original employed contexts.

In its simple form, a selection process filters *irrelevant* contexts using a heuristic based on a threshold. After the construction of a vector space and weighting, context elements that are associated with a weight or a frequency lower than a threshold are omitted from the vector space. The main assumption is that rare low-frequency context elements are uninformative and, therefore, do not influence the impending similarity assessments. For instance, in a text categorisation task that employs a document-by-term model, Yang and Pedersen (1997) show that statistical weight thresholding can be used reliably to halve the dimension of the vector space.

In a linguistic-entity-by-word model, a common selection process is to eliminate context words that belong to a *stop word list*. A stop word list is a fixed set of high-frequency words that are clearly not related to the devised semantic similarity application. Likewise, stemming and lemmatisation can be employed to reduce inflectional, and sometimes derivational, forms of words to a common base form. The experiments performed by Bullinaria and Levy (2012) suggest that although these techniques speed up the similarity computation by reducing the dimension of the vector space, they do not necessarily enhance the observed results. As described earlier in Section 2.2.3, linguistic information, such as syntactic relations, can also replace, or be combined with, statistical measures to select and filter contexts.

A selection process may also be used to rank and filter *redundant* contexts using an information theoretic/statistical measure. Information gain, mutual information, and $\chi^2$ test are examples of measures that can be used to check the correlations between context elements. If the correlation between context elements exceeds a certain threshold,

[1] Turney and Pantel (2010) also suggest that decreasing the sparsity will increase performance. However, they propose insufficient data as the major cause of the sparsity of vectors. Although insufficient data can contribute to the sparsity problem, one can hypothesise that the power-law distributions of contexts and entities play a more significant role in the sparsity of vectors than the data insufficiency. Further analysis is required to investigate the degree of the dimension expansion of a vector space against its sparsity reduction when the size of data increases.

one of them is considered to be redundant and can be eliminated from the list of employed contexts (see Hall, 1999, chap. 4 for further explanation). However, for a very high-dimensional vector space model that consists of hundreds of thousands of context elements, such methods are computationally inefficient.

In a more sophisticated approach, instead of a selection process, heuristics are used to implement a method of context generalisation. In Périnet and Hamon (2014b), context elements are generalised by finding synonym and hypernym-hyponym relationships between them. In the proposed, words in a sliding window constitute the context elements. To reduce dimensionality and sparseness of vectors, the context words are arranged into sets of words that are in a synonym or hypernym-hyponym relationship. To achieve the dimensionality reduction, the obtained sets replace context words (see also Périnet and Hamon, 2014a). Baker and McCallum (1998) uses a similar idea for dimensionality reduction in a document-by-term model in a text classification task. Baker and McCallum (1998) state that while this method enhances the result of the classification task in one corpus, it does not boost the performance in two other corpora. They conclude that the structure of data (e.g., the diversity of vocabulary, the distribution of words and the size of documents) plays a significant role in the performance of these methods of context generalisation.

The process described above leads to an alternative set of dimension reduction techniques known as *transformation* methods. A transformation method maps a constructed vector space $\mathbb{R}^n$ to $\mathbb{R}^m$ of lower dimensions—that is, $\tau : \mathbb{R}^n \mapsto \mathbb{R}^m, m \ll n$. The vector space at the reduced dimension $\mathbb{R}^m$ is the best approximation of the original model $\mathbb{R}^n$ in a sense. The approximation is evaluated by a criterion such as variance, gradient descent, or distance between context elements. The interpretation of these method using the distance between context elements in the transposed entity-context model is, perhaps, more compatible with the suggested mathematical perspective in this thesis. Based on the employed evaluation criteria, transformations are categorised as either *linear*, for example, truncated singular value decomposition, or *nonlinear*, for example, self-organising map.[1]

Truncated singular value decomposition (SVD) is the most familiar transformation-based dimensionality reduction technique in the vector space models of semantics (e.g., see Deerwester et al., 1990, the latent semantic analysis model (LSA)). Truncated SVD is a linear transformation method that exploits *the Euclidean norm* of context elements, or variance,[2] to compare a vector space with its projections in reduced dimensions. Given a vector space $\mathbb{R}^n$ consists of $p$ vectors, which is represented by a matrix $\mathbf{M}_{p\times n}$, the goal is to construct an $m$-dimensional vector space, represented by a matrix $\mathbf{M'}_{p\times m}$, $m \ll n$, that preserves most of the variance—thus, the Euclidean distances—in $\mathbf{M}$.

SVD factorises the matrix $\mathbf{M}_{p\times n}$ into the product of three matrices: $\mathbf{U}$, a $p \times p$ normalised orthogonal matrix (i.e., $\mathbf{U}\mathbf{U}^T = \mathbf{I}$); $\mathbf{\Sigma} = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$, a $p \times n$ diagonal matrix;

[1] Mathematically speaking, a selection process is a kind of linear transformation process.

[2] For a matrix $M_{p\times n}$, the Euclidean norm, also called the Frobenius norm, is defined as $\|M\|_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{n}|m_{ij}|^2}$.

and the transpose of an $n \times n$ normalised orthogonal matrix $\mathbf{V}$ (i.e., $\mathbf{V}\mathbf{V}^T = \mathbf{I}$):

$$\mathbf{M}_{p\times n} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \Big(\sum_{i=1}^{n} u_i \sigma_i v_i^T\Big)_{p\times n}. \tag{2.13}$$

The diagonal elements $\{\sigma_i\}$ of $\mathbf{\Sigma}$ are called the singular values of $\mathbf{M}$, and they are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$.[1] For a chosen $m$, $r \leq m \ll n$, the SVD truncation of $\mathbf{M}$ with rank $m$ is given by

$$\mathbf{M}'_{p\times m} = \mathbf{U}_{p\times m}\mathbf{\Sigma}_{m\times m}\mathbf{V}^T_{m\times m} = \Big(\sum_{i=1}^{m} u_i \sigma_i v_i^T\Big)_{p\times m}. \tag{2.14}$$

The basis elements of $\mathbf{M}'$ are orthogonal because the data is decorrelated in the $\ell_2$-norm (i.e., second-order) sense and thus their inner product is zero. According to Eckart and Young's (1936) theorem, $\mathbf{M}'$ represents the best approximation of $\mathbf{M}$ in $\mathbb{R}^m$, in which $\|\mathbf{M} - \mathbf{M}'\| = \sigma_{m+1}$ (see Martin and Porter, 2012, for references and elaboration).

The basis elements of the truncated vector space ($VSM_t$) that $\mathbf{M}'$ in Equation 2.14 represents express linear combinations of the correlated contexts in the original vector space ($VSM_o$) that $\mathbf{M}$ in Equation 2.13 represents. Therefore, in contrast to a selection process, the basis elements of $VSM_t$ cannot be directly labelled using the contexts employed in the $VSM_o$. Instead, they show *latent concepts* that express weighted combinations of contexts. Latent concepts may capture certain paradigmatic similarities, often called *high-order* structures, between the context elements employed in $VSM_o$ (see Leopold, 2005, for further mathematical explanation).[2]

Interpretation of the attached variances to context elements justifies different applications of truncated SVD. Turney and Pantel (2010) enumerate *latent meaning*, *high-order co-occurrence*, *sparsity reduction*, and *noise reduction* and leave the door open for further innovative applications. Under the assumption that the covariance of context elements indicates their similarity,[3] truncated SVD can be seen as a technique that exploits the Euclidean norm to measure similarity between context elements. Truncated SVD groups contexts into latent concepts such that it captures *latent meaning* and *high-order co-occurrences*; consequently, SVD truncation results in a vector space $VSM_t$ that expresses entities in a *latent semantic space*.

For instance, in the LSA model, a truncated SVD model represents the semantic relationships between documents using latent concepts that are derived from a document-by-word model. The latent concepts, also called *latent topics*, may capture synonymy relationships between words and enhance similarity measurements (Martin and Berry, 2011).[4] Consequently, the introduction of the latent concepts, which are more general than the contexts employed originally, results in the *sparsity reduction*. SVD truncation,

[1] $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}$ are called the principal components of $\mathbf{M}$.

[2] See also Sahlgren (2006, chap. 7) who suggests that the enhancements in TOEFL experiments with the LSA model are the result of encoding paradigmatic relations between context words using the truncated SVD.

[3] That is, the Euclidean distance between context elements in the transposed model.

[4] This argument can be generalised if synonymy relationship replaces *a paradigmatic relationship.*

however, does not guarantee generation of most suitable combinations of contexts for an intended application. For instance, in a cross-language information retrieval task performed on Wikipedia articles, Cimiano et al. (2009) report that truncated SVD does not enhance the obtained results.

Dimension reduction by truncated SVD implies that contexts associated with large variance express discriminative information. By the same token, under the Gaussian assumption of noise, the low variance contexts are supposed to be unimportant and noisy. Therefore, the truncation of SVD using highest singular values, as suggested in Equation 2.14, can be viewed as a filtering procedure that eliminates noise. The performance of noise reduction using SVD, however, depends on the distribution of the co-occurrences of linguistic entities and the context elements. While SVD truncation can be applied to remove Gaussian noise from data (e.g., white noise from sinusoidal signals), it fails with noise of a non-Gaussian nature. For instance, observations such as Figure 2.7a indicate that the co-occurrences of words in documents follow a non-Gaussian distribution (see also Sichel, 1975). Therefore, the use of SVD truncation for noise reduction is not effective in models that are based on the co-occurrences of words.

SVD is sensitive to the measurement scales of the context elements being analysed. Because a truncated SVD model retains linear combinations of the context elements that maximise the magnitude of variance, it is biased towards context elements that have larger variation values. If contexts are presented using values of different scales, then SVD truncation will be in the favour of context elements that are presented in scales of larger magnitude. Therefore, a scaling process is necessary before performing the SVD computation (see Jackson, 2004, for further information on methods of scaling).

In dimensionality reduction using the SVD truncation, the degree of dimension reduction should be decided by choosing a value for $m$ in Equation 2.14. An optimum value for $m$ is determined by the structure of the underlying data as well as the intended application. Direct selection of an optimum $m$, however, remains an open question (Martin and Berry, 2011). Therefore, the value of $m$ is often found by an exhaustive evaluation. In order to find the most satisfactory $m$, a performance measure suitable for the intended application is defined to compare several values of $m$. For example, in an information retrieval task, the estimated precision per $m$ in retrieval tasks decides the best degree of dimension reduction.

The computation of SVD for dimension reduction entails solving a linear equation that finds eigenvectors. For a given $n$-dimensional vector space, direct solution to this equation, known as the Gram–Schmidt process, is computationally trivial and of $O(n^2)$ complexity. Accordingly, the direct computation of truncated SVD for mapping $\mathbb{R}^n$ to $\mathbb{R}^m$, $m \ll n$, demands computational complexity proportional to $O(n^2 m)$. In practice, the singular values are approximated using iterative techniques such as the Lanczos method and its variations that take advantage of the sparseness of vector spaces (see Saad, 2003, chap. 7). In $k$ iterations, the $m$ largest singular values of a vector space are calculated directly and therefore the computational complexity of the transformation process is decreased to $O(nkm)$.

Truncated SVD requires the vector space of higher dimension than the targeted reduced dimension—that is, $\mathbf{M}$ in Equation 2.13—to be constructed prior to the process

of dimension reduction. However, this may not be desirable when dealing with large corpora. The size of a vector space that is built using a regular method of context matrix formation is a function of the size of the corpus. A regular context matrix formation associates entities to context elements, often using normalised values induced from the observed co-occurrence frequencies across the corpus. Such that context matrix becomes computationally intractable when the corpus size increases (e.g., Figure 2.7b). In a term-by-document model, for instance, the dimension of the vector space *dim* before the dimensionality reduction process is equal to the number of documents in the corpus $|c|$; appending $n$ new documents to the corpus corresponds to an increase in the dimension of the vector space—that is, $dim = |c| + n$. This is a non-trivial task when the corpus is big or its size increases at a sharp rate such as Web-scale information extraction tasks.

In addition to the aforementioned problems, the basis of the vector space with reduced dimensionality, which the data is projected onto, is also required to be devised prior to the projection task. If the structure of the data that is being analysed changes, the basis of the projected vector space also changes. Therefore, every time data is updated (i.e., a new context element or linguistic entity is added to the vector space), SVD should be recalculated in order to generate a suitable projection. This limitation is also generalised to dimensionality reduction techniques that are based on matrix factorisation techniques other than SVD, such as QR and ULV decomposition. *Random indexing* is an alternative dimension reduction technique that alleviates these issues.

The random indexing (RI) method, which is first introduced by Kanerva et al. (2000) for the construction of a word-by-document model and further delineated by Sahlgren (e.g., see Sahlgren, 2005, 2006), utilises all the advantages listed above to create a vector space model of semantics at reduced dimension. As recently described by QasemiZadeh and Handschuh (2015), the RI method can be seen in the form of a two-step procedure that consists of the construction of *a*) *index vectors* and *b*) *context vectors*. In the first step, each context is assigned to exactly one index vector $\vec{r}_{c_k}$. Sahlgren (2005) indicates that an index vector is a randomly generated high-dimensional vector, in which most of the elements are set to 0 and only a few to 1 and -1. In the second step, the construction of context vectors, each target entity is assigned to a vector of which all elements are zero and that has the same dimension as the index vectors. For each occurrence of an entity, which is represented by $\vec{v}_{e_i}$, in a context, which is represented by $\vec{r}_{c_k}$, the context vector for the entity is accumulated by the index vector of the context—that is, $\vec{v}_{e_i} = \vec{v}_{e_i} + \vec{r}_{c_k}$. The result is a vector space model, which is constructed directly at reduced dimension.

The procedure in the RI technique can be better explained by an example of a word-by-document model. In the first step of the process, each document in the corpus—that is, a context element—is assigned to an index vector $\vec{r}_{d_i}$ of dimension $m$ much smaller than $n$. Each word in the corpus is then assigned to an empty context vector $\vec{v}_{e_w}$—that is, all the elements of the vector are set to zero—and dimension $m$. The context vectors assigned to words can then be updated through a sequential scan of the corpus. For each occurrence of a word in a document $d_i$, its context vector $\vec{v}_{e_w}$ is updated such that $\vec{v}_{e_w} = \vec{v}_{e_w} + \vec{v}_{d_i}$. Given $n$ documents and $p$ words in the corpus, instead of a matrix $\mathbf{M}_{p\times n}$, the RI procedure results in a matrix $\mathbf{M}'_{p\times m}$ that represents the vector space model at reduced dimension by the factor $\frac{n}{m}$.

The random indexing method, thus, can be used to address a number of issues that are faced when using SVD truncation. For instance, in RI method, adding new context elements to the model is realised by adding new index vectors, without demanding a recalculation of the projection. Chapter 4 provides a comprehensive description and mathematical justification of the RI method. As is shown in Chapter 4, the RI method belongs to a category of dimensionality reduction techniques that are based on *random projections*.

The linear methods, such as SVD truncation and the RI method, have often been criticised for their inability to capture nonlinear structure of data beyond the $\ell_2$-norm (or, the second-order statistics). In contrast to linear techniques that assume the text data lies on a linear sub-space of a high-dimensional space, a number of dimensionality reduction techniques go beyond linearity assumption and explicitly reconstruct the data in an *embedded manifold*. These methods, known as nonlinear dimension reduction techniques, are further categorised by their underlying theory (e.g., see Van der Maaten et al., 2009, for a survey). In the context of natural language processing, Kohonen's (1990) self-organising maps is, perhaps, the most familiar example of a nonlinear dimensionality reduction technique (see also chapters of Honkela, 1997). Some experiments suggest that nonlinear methods do not necessarily outperform linear techniques, specially on real-world datasets containing noise or having discontinuous or multiple sub-manifolds (Huang and Yin, 2012).

While the use of neural networks and non-linear transformations are gaining popularity in several domains of study in distributional semantics, the study of these methods is left for another occasion. This section has only scratched the surface of the dimensionality reduction techniques that are most commonly applied in the distributional models of semantics. In the context of distributional models of semantics, dimension reduction techniques are still maturing with respect to several factors such as their performance, efficiency and underlying theories, as well as the data and intended applications of models. Figure 2.8 provides readers with a summary of the discussions in this section.

### 2.3.4 Similarity Measurement

The computation of vector similarities, which serves as a quantitative approximation of semantic relatedness between entities, is often the last step of the processes. As discussed in Section 2.2.1, a vector space model of semantics is endowed with structures called inner product, norm, and distance that are employed to define similarity measures between vectors. The cosine similarity and the Euclidean distance[1] are the familiar examples of such measures in the $\mathbb{E}^n$. Given the definition of inner product in $\mathbb{E}^n$ by Equation 2.8 and vectors $\vec{v}_i = \langle v_{i1}, v_{i2}, \cdots, v_{in} \rangle$ and $\vec{v}_j = \left\langle v_{j1}, v_{j2}, \cdots, v_{jn} \right\rangle$, the cosine similarity of $\vec{v}_i$ and $\vec{v}_j$ is given by the inner product of vectors when their length is normalised:

$$cos(\vec{v}_i, \vec{v}_j) = \frac{\langle \vec{v}_i, \vec{v}_j \rangle}{||\vec{v}_i||_2 ||\vec{v}_j||_2} = \frac{\sum_{k=1}^{n} v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^{n} v_{ik}^2 \sum_{k=1}^{n} v_{jk}^2}}. \tag{2.15}$$

[1] Also called 2-norm or $\ell_2$.

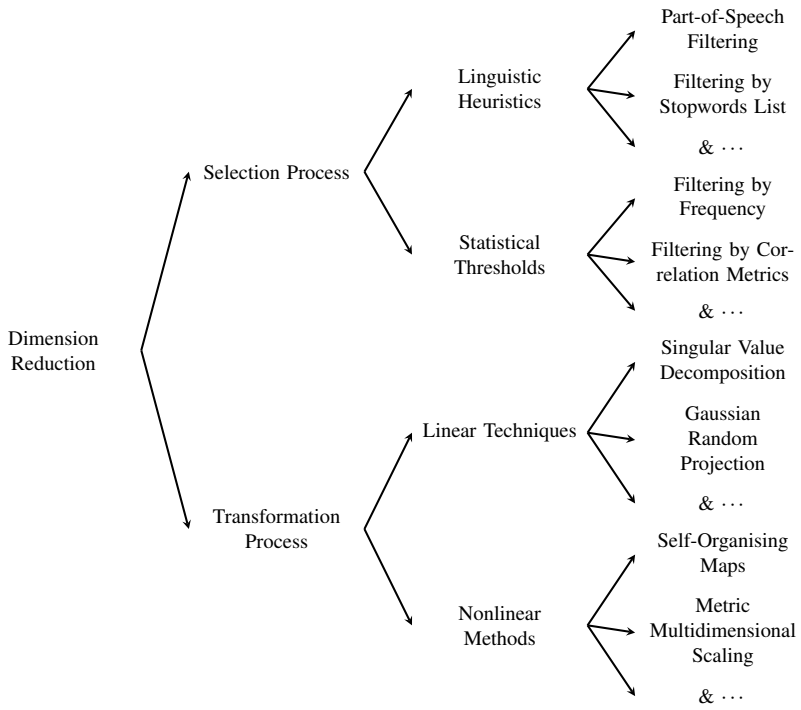



Figure 2.8: A map of dimensionality reduction techniques. Although not all methods neatly fall into the provided categorisation, it provides readers with a summary.

Likewise, the Euclidean distance is defined as:

$$d(\vec{v}_i, \vec{v}_j) = \|\vec{v}_i - \vec{v}_j\|_2 = \sqrt{\sum_{k=1}^{n}(v_{ik} - v_{jk})^2}. \tag{2.16}$$

As indicated by the numerator of Equation 2.15, the cosine similarity calculates the overlap between the vectors and thus it is a measure of the shared context elements between linguistic entities. In contrast, the Euclidean distance conveys the differences between corresponding context elements and thus it is a measure of discrepancy between linguistic entities.

The familiar Euclidean norm in a real vector space $\mathbb{R}^n$ can be replaced by other $p$-norms, $1 \leq p < \infty$,[1] in order to calculate similarity between vectors in $\ell_p$-normed spaces—that is, a vector space that is endowed with the $\ell_p$ norm.[2] For a given vector

[1] For $0 < p < 1$, the $p$-norm is called a quasi-norm, as it does not satisfy the triangle inequality in the definition of a norm. However, $\ell_0$—that is, $p = 0$—does not satisfy the homogeneity condition, and it is thus not a norm. From $\ell_0$ one can arrive at the definition of the Hamming distance. While Hamming spaces have been also used for similarity measurement in distributional semantics, their study goes beyond the scope of this thesis. A comprehensive study on the use of Hamming spaces in distributional semantics can be found De Vine (2013) and De Vries (2014) (see also Gionis et al., 1999).

[2] Remember from Section 2.2.1 that $\mathbb{E}^n$ is a $\mathbb{R}^n$ that is endowed with the Euclidean norm (i.e., the $\ell_2$-norm); it is thus an $\ell_2$-*normed space*.

| Name | Formula |
|---|---|
| Dice | $s_{\text{Dice}}(\vec{v}_i, \vec{v}_j) = \frac{2\sum_{k=1}^{n} v_{ik} v_{jk}}{\sum_{k=1}^{n} v_{ik}^2 + \sum_{k=1}^{n} v_{jk}^2}$ |
| The harmonic mean | $s_{\text{HM}}(\vec{v}_i, \vec{v}_j) = 2\sum_{k=1}^{n} \frac{v_{ik} v_{jk}}{v_{ik} + v_{jk}}$ |
| Jaccard | $s_{\text{Jaccard}}(\vec{v}_i, \vec{v}_j) = \frac{\sum_{k=1}^{n} v_{ik} v_{jk}}{\sum_{k=1}^{n} v_{ik}^2 + \sum_{k=1}^{n} v_{jk}^2 - \sum_{k=1}^{n} v_{ik} v_{jk}}$ |

Table 2.3: Examples of similarity measures in the inner product family. In these equations, similar to the cosine similarity in Equation 2.15, the inner product of vectors in the denominators of the formulas is normalised using different values. These measures show the commonality between vectors.

$\vec{v}$ in an $\ell_p$-normed space, the Euclidean norm $\|\vec{v}\|_2$ in Equation 2.8 is generalised to

$$\|\vec{v}\|_p = \Big(\sum_{i=1}^{n} |v_i|^p\Big)^{\frac{1}{p}}. \tag{2.17}$$

Hence, the distance between the two vectors $\vec{v}_i$ and $\vec{v}_j$ in a $\ell_p$-normed space—also known as the Minkowski distance—is given by

$$d_p(\vec{v}_i, \vec{v}_j) = \|\vec{v}_i - \vec{v}_j\|_p = \sqrt[p]{\sum_{k=1}^{n} |v_{ik} - v_{jk}|^p}. \tag{2.18}$$

Amongst the $d_p$ distances, besides the Euclidean distance, the $\ell_1$ distance, also known as the Manhattan distance or city block distance, has been employed for semantic similarity measurement.

As discussed earlier, the collected frequencies of the co-occurrences of linguistic entities and context elements can be interpreted in mathematical frameworks other than the vector space model. Therefore, it is common to employ probabilistic and information-theoretic measures for similarity calculation. Many of these measures satisfy the axioms listed in the definition of distance (norm)[1] and therefore can be categorised in an $\ell_p$ distance family. From this perspective, a $d_p$ distance can be normalised in different ways to design new distance measures. However, there are many other measures that do not satisfy the required axioms for a distance metric. An example of this categorisation is given by Cha (2007).

Cha provides a survey of similarity measures and their properties. He enumerates dozens of similarity measures and groups them according to their syntactic characteristics (i.e., the homogeneity of their formulas), the correlation between their generated results in a clustering task, and the caveats in their implementations. Following his survey, Tables 2.3, 2.4, and 2.5 provide a list of similarity measures analogous to $\ell_1$, $\ell_2$, and inner product

[1] See on page 32.

| Name | Formula |
|---|---|
| Bray-Curtis | $s_{\text{BC}}(\vec{v}_i, \vec{v}_j) = \frac{\sum_{k=1}^{n} \lvert v_{ik} - v_{jk} \rvert}{\sum_{k=1}^{n} v_{ik} + v_{jk}}$ |
| Canberra | $s_{\text{Can}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} \frac{\lvert v_{ik} - v_{jk} \rvert}{\lvert v_{ik} \rvert + \lvert v_{jk} \rvert}$ |
| Gower (see Pavoine et al., 2009, for description) | $s_{\text{Gower}}(\vec{v}_i, \vec{v}_j) = \frac{1}{k} \sum_{k=1}^{n} \frac{\lvert v_{ik} - v_{jk} \rvert}{w_k}$ |
| Soergel | $s_{\text{Soe}}(\vec{v}_i, \vec{v}_j) = \frac{\sum_{k=1}^{n} \lvert v_{ik} - v_{jk} \rvert}{\sum_{k=1}^{n} \max(v_{ik}, v_{jk})}$ |

Table 2.4: Examples of (dis)similarity measures in the $\ell_1$ distance family. In the definition given for $s_{\text{Gower}}$, $w_k$ indicates the range of the values for the $k$th element of vectors.

| Name | Formula |
|---|---|
| Clark | $s_{\text{Clark}}(\vec{v}_i, \vec{v}_j) = \sqrt{\sum_{k=1}^{n} \left( \frac{v_{ik} - v_{jk}}{v_{ik} + v_{jk}} \right)^2}$ |
| Symmetric $\chi^2$ | $s_{Sym_{\chi^2}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} \frac{(v_{ik} - v_{jk})^2}{\max(v_{ik}, v_{jk})}$ |
| Weighted Euclidean | $s_{\text{WE}}(\vec{v}_i, \vec{v}_j) = \sqrt{\sum_{k=1}^{n} \frac{(v_{ik} - v_{jk})^2}{w_k}}$ |

Table 2.5: Examples of (dis)similarity measures in the $\ell_2$ distance family. In the definition given for $s_{\text{WE}}$, $w_j$ denotes a weighting value.

formula, respectively (to verify the given definitions, see Deza and Deza, 2006, 2014). Examples of information-theoretic similarity measures are given in Table 2.6.

Amongst his observations, Cha suggests that the family of inner product measures, such as cosine, generates results closely related to $\ell_2$ distance. In addition, the results generated by the two distance metrics $d_a$ and $d_b$ are highly correlated if $d_a = cd_b$ or $d_a = 1 - d_b$. Particularly, in distributional semantics, because of the sparseness of vectors, a method of *smoothing* is required to alleviate these problems, which is a major research problem on its own (e.g., see Chen and Goodman, 1999). For example, in these cases, one solution is to replace zero with a very small value—that is, the additive smoothing technique.

There is an extensive body of research on learning distance metrics, with detailed studies that go beyond the scope of the discussion in this section. In these methods, a distance metric is altered, often using a weight normalisation mechanism in order to reflect a set of given constraints on similarities (e.g., $w_k$ in the definition of $s_{\text{WE}}$[1] in Table 2.5 and $s_{\text{Gower}}$ in Table 2.4). The weight normalisation problem is usually modelled as a learning

[1] In this context often called the Mahalanobis distance.

| Name | Formula |
|---|---|
| Bhattacharyya | $s_{\mathrm{B}}(\vec{v}_i, \vec{v}_j) = -\ln \sum_{k=1}^{n} \sqrt{v_{ik} v_{jk}}$ |
| Hellinger | $s_{\mathrm{H}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} (\sqrt{v_{ik}} - \sqrt{v_{jk}})^2$ |
| K-Divergence | $s_{\mathrm{KD}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} v_{ik} \ln(\frac{2v_{ik}}{v_{ik}+v_{jk}})$ |
| Kullback-Leibler | $s_{\mathrm{KL}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} v_{ik} \ln(\frac{v_{ik}}{v_{jk}})$ |

Table 2.6: Examples of information theoretic similarity measures adopted in the vector space models, assuming vectors represent probabilities.

task in the framework of an optimisation problem. For instance, given constraints in the form of '$x$ is close to $y$' for a set of pairs of vectors $x$ and $y$, Xing et al. (2002) suggest a method that learns a distance metric. Schultz and Joachims (2004) suggest a similar technique, however, when the constraints are given in the form of a set of triplets such as '$x$ is closer to $y$ than it is to $z$'. In the machine learning literature, metric learning is often studied as a learning scheme for feature weighting (see Kulis, 2013, for survey and references). These techniques, thus, can be perceived in combination with the preceding weighting step in which more indicative contexts are assigned to higher weights in order to increase their impact on the similarity measure. Alternatively, given a known set of related vectors, it is possible to compare distance metrics in order to choose the most suitable one.

Bullinaria and Levy (2007) provide a comparison between several similarity measures. The comparison is carried out by studying the results of four different experiments that employ word-by-word models:

**TOEFL** : From four given choices, a word is selected that has the closest meaning to a target word in a dataset consisting of 80 questions.

**Distance** : Similar to the TOEFL test, but the distance between a pair of semantically related words (e.g., lettuce and cabbage) is compared with the distances between 10 randomly chosen pairs of words from a set of 200 words in order to assess the structure of the model at a larger scale than the TOEFL test.

**Syntactic Clustering** : The distance between a target word's vector and the centre of a cluster that represents its syntactic category is measured and the ratio of words that are closer to their real syntactic category than another is defined as the performance measure. The test is limited to 100 words from 12 different syntactic categories.

**Semantic Clustering** : The same test as above, however, for semantic categories. The performance measure is defined as the ratio of words that are closer to their own semantic category than others. The experiment is limited to 530 words in 53 semantic categories.

| Similarity Measure: Rank | Experiment (TOEFL) | Experiment (Distance) | Experiment (Synt. Cluster) | Experiment (Sem. Cluster) |
|---|---|---|---|---|
| 1 | Hellinger | Kullback-Leibler | City Block | Kullback-Leibler |
| 2 | Bhattacharya | City Block | Hellinger | Hellinger |
| 3 | City Block | Hellinger | Bhattacharya | Bhattacharya |
| 4 | Kullback-Leibler | Bhattacharya | Cosine | City Block |
| 5 | Cosine | Cosine | Kullback-Leibler | Cosine |
| 6 | Euclidean | Euclidean | Euclidean | Euclidean |

Table 2.7: Performance of similarity measurements with respect to each other in Bullinaria and Levy's (2007) experiments; rank 1 shows the best-performing similarity measure.

| | Experiment (TOEFL) | Experiment (Distance) | Experiment (Synt. Cluster) | Experiment (Sem. Cluster) |
|---|---|---|---|---|
| **Best** $\approx$ % | 75 | 90 | 92 | 71 |
| **Worst** $\approx$ % | 65 | 85 | 82 | 58 |

Table 2.8: Approximate values for the best and the worst performances of similarity measurements in Bullinaria and Levy's (2007) experiments.

Table 2.7 represents the performance of the similarity measures in the tasks explained above. The results shown in the table are limited to when vectors are weighted such that they represent the conditional probabilities $p(w_c|w_t)$, where $w_t$ and $w_c$ are the target and context word, respectively. As is shown in the table, the best performing measure varies from one experiment to another. While a similarity measure such as city block has a constant superior performance with respect to measures such as the Euclidean and the Cosine, this relationship does not hold for other metrics such as the Kullback-Leibler and Hellinger. An approximate difference between the best and worst performing measures is shown in Table 2.8. As suggested in Cha (2007), the Kullback-Leibler and Hellinger, and the Cosine and Euclidean show similar behaviours in Bullinaria and Levy's (2007) experiments.

In an earlier experiment similar to Bullinaria and Levy's (2007) *Distance* test, Lee (2001) provides a report of the performance of similarity measures which is analogous to the result shown in Table 2.7. She also reports that the city block outperforms the cosine, and the cosine outperforms the Euclidean distance, whereas a weighted Kullback-Leibler method, called *skew divergence*, gives the best performance. However, Bullinaria and Levy (2007) show that the cosine similarity can outperform all the similarity measures in every one of the above tasks when a suitable weighting process, such as *pointwise mutual information*, substitutes the probability weighting. In another experimental setup, Curran (2004, chap. 4) suggests that the Dice and Jaccard outperform the cosine similarity measure. In an automatic synonym acquisition task, Shimizu et al. (2008) report

that a weighted Euclidean measure, which obtains weights through a supervised learning method, outperforms all other metrics in their experiment. In their reported experiment, the cosine and the Jaccard are the next-best-performing measures and are listed above the Euclidean and, contrary to the above reported-experiments, the city block measure.

In an alternative approach, instead of mere performance comparison, Weeds et al. (2004) suggest an attributive comparison of similarity measures. In a synonym detection task, Weeds et al. (2004) compare 10 various similarity measures by investigating the frequency characteristics of target words and their closest neighbour words given by a similarity measure. They correlate the frequency of the obtained neighbour words to their *distributional* and *semantic generality* and accordingly classify similarity measures into three groups. The first group of measures are those that are biased towards selecting high-frequency, and thus more general, words. The second group of measures are those that are more sensitive to to low-frequency, thus more specific, words. The third group consists of those measures that are in favour of with a similar frequency to target words. In their experiment, the cosine and the skew divergence are categorised in the first group, whereas the Jaccard and the harmonic mean are classified in the third group. A similar study of similarity measures in an information retrieval context is given in Jones and Furnas (1987).

Mathematically speaking, the distribution pattern of entities in a vector space determines the performance of similarity measures. In the absence of *a priori* knowledge of the distribution of data, similarity measures are often evaluated empirically. An approach, such as that described above or proposed by Lin (1998b), is employed to interpret similarity measures' performance and elucidate their differences. With such intuition, as an example, Lee (1999) suggests that for sparse models, commonality-based similarity measures—such as the Dice and the cosine—are expected to outperform those that are based on differences such as the Euclidean distance. In information retrieval, Jones and Furnas (1987) compare the sensitivity of several similarity measures to *within-object* and *between-object* differences and conclude in favour of the cosine measure.

The literature reviewed unanimously agrees that various similarity measures exhibit different behaviours in different tasks and thus there is no single superior measure for all applications. In a given application, therefore, the choice of a similarity metric is likely to affect the quality of the observed result.

### 2.3.5 Orchestrating the Processes

This section concludes our discussion on the processes in vector space models of semantics by emphasising the importance of a holistic approach to their design and implementation. As described, the goal of the chain of processes introduced in this section is to simulate a sense of semantic relatedness between vectors that represent the linguistic entities being modelled. As is explained, the semantic relatedness is ultimately translated into the proximity of vectors, which is transpired by a notion of similarity measure. The efficacy of measures is predominated by the distribution of vectors, which, in turn, is a function of the answer to the earlier question of 'what the context elements are'. A change in context elements results in the transformation of the vectors' distribution in the model

and thus it is highly likely to cause redesign in the subsequent processes, amongst them the similarity measurement.

Usually, the use of one specific method in one of the processes introduced in this section limits the choice of methods that are available to be applied in the remaining processes. For instance, the choice of a random projection with Gaussian random matrix for the dimensionality reduction limits the options for the similarity measurement. Similarly, the choice of random indexing limits the options for the weighting process. As discussed in Chapter 4, using random indexing for collecting co-occurrences results in a Euclidean vector space model; therefore, the use of similarity measures other than the $\ell_2$ distance family cannot be justified, at least mathematically. In other words if for any reason, the use of norms other than $\ell_2$ is preferable, then a Gaussian random projection technique such as random indexing cannot be employed. With the same rationale, using SVD truncation is not justified when similarities are measured using a metric other than the $\ell_2$ distance family.

Moreover, one method can neutralise the advantages of another method. For example, normalising the Euclidean distance by the inverse of the variance of contexts in a vector space model that is induced by SVD truncation has no effect on the obtained similarities. In the same way, if SVD truncation is used for the dimensionality reduction, a weight scaling is recommended as a pre-processing step. Nonetheless to say that Likewise, a number of similarity measures, such as the familiar Euclidean and cosine similarity, are equivalent if the vectors are normalised to unit length. In contrast, as the experiment shows, the right combination of methods in the above processes can enhance the observed results dramatically.

Last but not least, the suggested cascaded architecture for processes, in which one process is applied after the other in a pipeline, may not be applicable or desirable in a real-world application. The suggested arrangement of the processes and the clear-cut boundaries between them are given solely for clarity in the presentation. The software architecture of an implemented distributional semantic method may require a complex sequence of interactions far beyond what is described in this section.

## 2.4 Classification in Vector Spaces

In a vector space model of semantics (VSM), a variety of machine learning algorithms can be employed to address a range of classification and clustering problems. A class is a set of entities that can be identified by characteristics that all its members share. The classification problem is the task of automatic assignment of entities to classes. However, if the classes are not known prior to the assignment task, then the task is called clustering. Clustering thus is the task of grouping entities by their mutual characteristics in such a way that the members of a group, called a cluster, are more similar to each other than to the members of other clusters in a sense. The classification task is usually referred to as *supervised learning*, whereas the clustering task is known as *unsupervised learning*.

Familiar examples of such tasks are document classification and clustering. In a document-by-term model, instead of measuring similarities between a pair of documents, or a query and a document, the documents are categorised by certain criteria, for instance,

their subject areas. In this example, if the subject areas are known beforehand—for example, the subject areas are limited to science and art—the task is called document classification. However, if the subject areas are not known beforehand, then the task is called document clustering and it organises the documents, for this given example, into groups that give a sense of the subject areas. Using different context types, documents can be classified, instead of by subjects area, by their relatedness, style, theme, sentiment, author characteristics, etc.

In the combination of a learning technique with a vector space model, the learning algorithm compares the vectors by its own implemented logic of similarity. In a vector space model, which interprets the meaning of linguistic entities such as documents using the geometry of vectors, a class or a cluster refers to a collection of vectors that form a region. A learning algorithm consequently identifies these regions. This perception implies the assumption that entities of the same class or cluster form a *contiguous* region and regions of different classes do not overlap.[1] Violations of these assumptions are the main causes of inaccuracy in classification and clustering tasks.[2]

A classification task—that is, supervised learning—can be formalised by a mapping function $f$. For a vector space $V$ and an output space $L$, which consists of a finite set of category labels $l$, the classification process is given by $f : V \mapsto L$. The mapping function $f$ is *learned* by a machine learning algorithm during a process called *training*. The training process chooses a function that *best* estimates the relationship between the input vectors and the output labels from a given set of instances $T \in V \times L$, which is called the *training dataset*. If $L = \mathbb{R}$, then the classification task is called regression. For $|L| = 2$, the task is called *binary* classification. If $|L| > 2$, then the task is called *multi-class* or *multi-way* classification. In a clustering task—that is, unsupervised learning—the $T$ and $L$ are not presented explicitly. Instead, criteria—such as the cardinality of $L$, the way similarities are compared, and a relationship between members of clusters—are given.

These learning algorithms are the subject of vibrant scientific research in a framework known as *statistical learning theory*. The comprehensive study of these methods, therefore, requires dedicated research. In this section, however, the surface of topics in statistical learning theory are scratched and only learning methods that take a geometric approach to a classification task are introduced. These methods classify data in a normed space and, thus, are compatible with the interpretation principles of vector space models, which are introduced earlier in Section 2.2. The methods introduced in this section are used later in this thesis.

In statistical learning theory, learning procedure is formalised using a mapping function $(V \times L)^n \mapsto \mathcal{F}$. In this definition, $\mathcal{F}$, which is called the hypothesis space, is a space of functions $f_m : V \mapsto L$, where $V$ and $L$ are the input vector space and the output label space, respectively. The learning algorithm searches in $\mathcal{F}$ for a function that best approximates the relationship implied between the vectors and the labels by the set of $n$ samples from $(V \times L)^n$. This formalisation is based on two assumptions. First, it is assumed that the data is being classified, that is, the set of $n$ tuples $\langle \vec{v}, l \rangle$, are drawn independently and

[1] Evidently, it can be also interpreted as a corollary to the distributional hypothesis.

[2] Alternatively, in a probabilistic framework, classes are interpreted as hidden properties of entities, often named latent variables.

identically from a fixed but unknown joint probability distribution $p(\vec{v}, l)$. Second, in order to assess the quality of learning, it is assumed that there is a notion of *loss* or error that can determine, for a given input vector, the discrepancy between the expected label and the label predicted by a $f_m$. This is indicated by a *Loss function loss* : $L \times L \mapsto \mathbb{R}$. For a given vector $\vec{v}$ and the expected label $l$, $Loss(l, f_m(\vec{v}))$ gives the error of $f_m$.

By these assumptions, the goal of the learning process is to find a $f_o \in \mathcal{F}$ that minimises the average error. For $f \in \mathcal{F}$, the average error, which is also called the *risk* of $f$ $R(f)$is given by:

$$R(f) = \int_{V \times L} Loss(l, f(\vec{v}))\mathrm{d}p(\vec{v}, l). \tag{2.19}$$

However, $R(f)$ cannot be computed because the probability distribution $p(\vec{v}, l)$ is unknown. The learning problem formalised above can be solved using a variety of approaches. From one perspective, similar to the proposed taxonomy of the distributional methods in Section 2.1.2,[1] the learning techniques can be categorised into methods that provide a solution using probability estimation techniques or methods that interpret the learning problem in a metric space.[2] As cited by Jain et al. (2000), however, under certain assumptions on the probability distributions, the two approaches are equivalent.

In the probability-based category, two major approaches to approximate $R(f)$ can be recognised. In the first group of methods, it is assumed that the type of the distribution of data is known; thus, a probability model with a number of fixed parameters can be used to estimate $p(\vec{v}, l)$. Consequently, the training dataset $T$ is used to estimate the value of the model's parameters. For instance, assuming the data has a Gaussian distribution, the joint probability is estimated using the mean and variance of the data samples in $T$. The familiar algorithm in this group is the naïve Bayes classifier.

The second group of probability-based methods, in contrast to the former methods, do not assume prior knowledge of the type of data distribution. These techniques estimate $p(\vec{v}, l)$ by the observation of the data samples provided in $T$. In distributional semantics, the Blei et al.'s (2003) latent Dirichlet allocation for uncovering *topic models* is a well-known example of these methods. Both category of methods listed above can exploit the learned joint distribution in a reverse fashion; that is, given a class label $l$, they can synthesise examples of context elements related to $l$. Hence, the probability-based methods are often known as *generative* approaches.

On the other side, one category of learning techniques—often named as *discriminative* methods—bypasses the probability estimation and approximates $R(f)$ directly. A subcategory of these methods adopt a *geometric approach* in the sense that they reformulate a learning task as the construction of decision boundaries in a metric space. The support vector machine algorithm and the $k$-nearest-neighbours technique are the familiar examples in this category. These methods approximate $R(f)$ from the training set $T$ using an *induction principle* such as *empirical risk minimisation* (ERM). Given $n$ samples $\langle \vec{v}_i, l_i \rangle$

[1] See Figure 2.2.

[2] This inventory can be expanded, for example, by adding information-theoretic-based approaches, etc.

in $T$, the *empirical risk of function* $f$ over $T$ is given by:

$$R_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n} Loss(f(\mathbf{v}_i), l_i). \tag{2.20}$$

It is expected that the function $f$ that has a small empirical risk (i.e., $R_{\text{emp}}(f)$) will also have a small risk (i.e., $R(f)$). It is proved that for $f$ of *finite complexity*, $R_{\text{emp}}(f)$ converges to $R(f)$ when $n \to \infty$ (see Evgeniou et al., 1999, for further explanation). Therefore, it is assumed that the goal of a learning task can be achieved—that is, finding the $f_o \in \mathcal{F}$ that minimises the risk $R(f)$—by finding the $f_o$ that minimises the empirical risk $R_{\text{emp}}(f)$:

$$f_o = \underset{f\in\mathcal{F}}{\text{argmin}}\, R_{\text{emp}}(f) = \underset{f\in\mathcal{F}}{\text{argmin}}(\frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{v}_i), l_i)). \tag{2.21}$$

Accordingly, $R_{\text{emp}}(f)$ is employed as a quantifiable method for the assessment of the *generalisation* ability of $f_o$—that is, it is assumed that if $f_o$ has a small $R_{\text{emp}}(f)$, then it also has a high generalisation ability.[1] Whereas research in machine learning investigates developing algorithms by suggesting induction principles other than ERM,[2] and imposing restriction on the complexity of $\mathcal{F}$[3], in this thesis, the scope is limited to the use of memory-based $k$-nearest neighbours ($k$-nn) algorithms. The $k$-nn algorithm implies that the $f_o$ that determines class labels by taking an average of the class labels of instances in $T$ that are close to input $\vec{v}$ has the lowest $R_{\text{emp}}$.[4]

### 2.4.1 The *k*-Nearest Neighbours Algorithm

The $k$-nearest neighbours ($k$-nn) algorithm is a learning technique that is explained by the geometry of vectors in space (Cover and Hart, 1967).[5] In $k$-nn, instances of data—that is, vectors—are classified based on the class of their nearest neighbours. It is a two-step process: in the first step, the $k$ closest vectors to the data item being classified are located; in the second step, the class label of the data item is determined using the class label of these nearest neighbours.

Given a vector space $V$ and a training dataset $T \in V \times L$, where $L$ is a finite set of class labels, it is assumed that there exists a distance function $d : V \times V \to \mathbb{R}$, such as that given in Section 2.3.4, that assigns a distance value $d(\vec{v}, \vec{t})$ to each pair of vectors $\vec{v} \in V$ and $\vec{t} \in T$. In its simplest form, when $k = 1$, for an input vector $\vec{v} \in V$, $T$ is searched for the $\vec{t}$

[1] Although in real-world applications, this assumption does not hold. If the training dataset is small or the hypothesis space $\mathcal{F}$ is large, then there are many functions that can satisfy Equation 2.21. Under these conditions, however, using ERM may not necessarily result in a function that has a high generalisation ability. Under such circumstances, a function $f_o$ that shows a high performance during the learning procedure shows a poor performance when dealing with data samples other than $T$. This is often called *overfitting*.

[2] Which its study goes beyond the scope of this thesis.

[3] For example, using the assumption that the target function $f_o$ is in the form of a *linear discriminant function*.

[4] Also, see Kulkarni and Harman (2011), for further elaboration of statistical learning theory and stimulating questions.

[5] Perhaps more intuitive than SVM.

that has the least distance to the $\vec{v}$ and its class label is assigned to the $\vec{v}$. This classification task can be formalised by the mapping function *nn* that returns corresponding label $l \in L$ of vector $\vec{t}$ such that:

$$nn(\vec{v}) = l_{\vec{t}}, \text{ where } \vec{t} = \underset{\vec{y} \in T}{\operatorname{argmin}}\, d(\vec{v}, \vec{y}). \tag{2.22}$$

By the same token, the $nn(\vec{v})$ can be generalised to $k$ neighbours. After finding the $k$ closest instances in $T$ to $\vec{v}$, that is $\{t_1 \cdots t_k\}$, the most straightforward approach—known as *unweighted voting*—is to assign the majority class label among the $k$ nearest neighbours to the data item being classified:

$$k\text{-}nn(\vec{v}) = l_y, \text{ where } l_y = \underset{l \in L}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(l, f(\vec{t_i})), \tag{2.23}$$

where $f(\vec{t_i})$ denotes the class label of $\vec{t_i} \in T$, and $\delta(x, y)$ is a function that compares the two class labels $x$ and $y$, that is:

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}. \tag{2.24}$$

However, a *distance weighted* method can replace the unweighted sum of labels:

$$k\text{-}nn(\vec{v}) = l_y, \text{ where } l_y = \underset{l \in L}{\operatorname{argmax}} \sum_{i}^{k} w_i \delta(l, f(\vec{t_i})), \tag{2.25}$$

where $w_i$ is real valued function on the distance between $\vec{v}$ and instances from the training set. For example, the weight function can be defined as an inverse of the distances between $\vec{v}$ and $\vec{t_i} \in T$, that is:

$$w_i = \begin{cases} 1 & x = y \\ \frac{1}{d(\vec{v}, \vec{t_i})} & x \neq y \end{cases}. \tag{2.26}$$

Similarly, as suggested by Daelemans et al. (2009) and Cunningham and Delany (2007), $w_i$ can be defined using an exponential function based on Shepard's (1987) justification, that is:

$$w_i = e^{-\alpha d(\vec{v}, \vec{t_i})^{\beta}}, \tag{2.27}$$

where $\alpha$ and $\beta$ are constant, often $\alpha, \beta = 1$, that are used to control the power of exponential decay factor. The $k$-nn algorithm, thus, can be alternated by adopting different approaches for assigning class labels through definitions of $\delta$ and $w$.

The $k$-nn algorithm is known to be a *lazy-learning* technique, which means that it does not require a training procedure prior to the classification task. The induction takes place during run-time and using training data samples that are presented explicitly. The main computation in the learning and classification task is the scoring of training vectors against an input vector in order to find the $k$ nearest neighbours. The $k$-nn, therefore, is also known as an *example-based* or *case-based* learning technique. It is a simple yet effective method of classification that has been widely used in many applications.

However, the application of $k$-nn requires selecting the $k$ value where it is dependent on the distribution of the data is being classified, the distribution of training samples, and the metric that is used to find the nearest neighbours. The value for $k$ is usually selected by a heuristic technique such as cross-validation. In general, larger values of $k$ are believed to reduce the effect of noise; however, this makes class boundaries less distinct. For small values of $k$, the $k$-nn method is also known to be sensitive to the presence of noisy or irrelevant data (Yang, 1999). In addition, when the number of training instances increases, the performance of $k$-nn reduces. However, these limitations have been actively addressed by a large number of research.

Besides the mathematical account given above, based on the application's context, there are several interpretations of the $k$-nn algorithm. In its simplest form, $k$-nn can be seen as a ranking system in which a threshold is used for assigning a class label to an input vector (e.g., Bustos and Navarro, 2004). In the context of distributional semantics, however, the $k$-nn algorithm can be best explained by the substantial research efforts that are often flagged by the term *memory-based language processing* (Daelemans and van den Bosch, 2005)—that is, as described by Daelemans (1999), a union of the two tradition of analogy-based language models in linguistics, and $k$-nn learning technique in artificial intelligence.

As summarised in Daelemans and van den Bosch (2010), $k$-nn can be seen as a *similarity-based reasoning* process in which the learning process is analogous to memorising (i.e., storing) a set of examples. Whereas a number of learning techniques employ a meta-language such as rules to construct an abstract representation of text data (known as *eager* learning methods), $k$-nn relies directly on the text data to perform the classification task. Hence, similar to the discussion in Chapter 1, $k$-nn offers an empiricist method of classification. Training text samples are, thus, can be kept in their original format with no alteration. As a result, it can be suggested that:

- the process of classification in $k$-nn is more intuitive than methods that use an abstract representation of the training data;
- language exceptions and less frequent patterns, which are often ignored by a generalised representation of the training data, can be handled effectively;
- even using a very small set of training examples, $k$-nn shows a reasonable generalisation ability.

In the context of this thesis, the $k$-nn method is employed for two of its particular characteristics:

- its plausible compatibility with the distributional hypothesis and its intuitive explanation of the classification task;
- its memory-based learning strategy.

As explained above, the former characteristic introduces $k$-nn as a cognitively plausible data-driven approach for similarity-based reasoning, whereas the second characteristics make it exceptionally flexible and suitable for implementing an interactive learning algorithm. No training process is required to develop a model and the examples can be

added or removed at anytime during the deployment of the method. Hence, the memory-based learning is a simple yet effective approach for the iterative development of terminological resources—in which the model can be updated as a user annotates and organises terms. Lastly, the example-based classification method can be easily scaled out, for example, with the help of *MapReduce programming model*—which is an important feature in big text data analytics.

## 2.5 Chapter Summary

The discussion in this chapter started by giving an overview of the distributional hypothesis and the vector space models of semantics, which form the theoretical basis for the proposed methods in this thesis (i.e., Section 2.1). Vector spaces as an algebraic structure are described in Section 2.2.1; Section 2.2.2 explained how these algebraic structures are employed to model and interpret distributional properties of linguistic entities in various contexts in order to capture meanings. In Section 2.2.3, this discussion was accompanied by a survey of the employed context elements and types of semantic models that have been employed in different text processing tasks; for example, to address problems in applications such as information extraction and retrieval.

Processes in vector space models of semantics were a major part of the discussion in this chapter (i.e., Section 2.3). The steps that are necessary to build a vector space model are reviewed. These processes, from the vector space construction to the similarity measurement process, were discussed in detail. Accordingly, Section 2.4 explained the use of learning techniques in distributional semantic models, in which an emphasis was put on the methods that employ the geometry of vectors in order to perform a classification task. Particularly, in Section 2.4.1, the $k$-nearest neighbours algorithm, which will be employed later in this thesis, was introduced.

# Reference List


Anderson, A. J., Bruni, E., Bordignon, U., Poesio, M., and Baroni, M. (2013). Of words, eyes and brains: Correlating image-basedased distributional semantic models with neural representations of concepts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1960–1970, Seattle, USA. Association for Computational Linguistics. 40

Anderson, A. J., Bruni, E., Uijlings, J., Bordignon, U., Baroni, M., and Poesio, M. (2012). Representational similarity between brain activity elicited by concrete nouns and image based semantic models. In *Proceedings of the Workshop on Vision and Language (VL'12)*, University of Sheffield, UK. 39

Angeli, G. and Manning, C. D. (2014). NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–545, Doha, Qatar. Association for Computational Linguistics. 39

Baker, L. D. and McCallum, A. K. (1998). Distributional clustering of words for text classification. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, Australia. ACM. 47

Baroni, M. (2013). Dr. Strangestats or: How I learned to stop worrying and love distributional semantics. *Computational Models of Language Meaning in Context (Dagstuhl Seminar 13462)*, 3(11):85–86. 28

Baroni, M. and Evert, S. (2009). Statistical methods for corpus exploitation. In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics: An International Handbook*, volume 2 of *Handbooks of Linguistics and Communication Science*, pages 777–802. Walter de Gruyter. 24

Baroni, M., Lenci, A., and Onnis, L. (2007). ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics. 44

Baroni, M., Murphy, B., Barbu, E., and Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34(2):222–254. 36, 37, 38, 40

Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Van der Vorst, H. (1993). *Templates for the solution of linear systems: Building blocks for iterative methods*. SIAM. 42

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor"meaningful? In Beeri, C. and Buneman, P., editors, *Database Theory – ICDT'99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin Heidelberg. 45

Bins, J. and Draper, B. (2001). Feature selection from huge feature sets. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, volume 2, pages 159–165, British Columbia, Canada. 44

Blackburn, P. and Bos, J. (2005). *Representation and inference for natural language: A first course in computational semantics*. CLSI Studies in Computaional Linguistics. CSLI. 23

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022. 29, 60

Bruni, E., Tran, N. K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47. 39

Bruni, E., Uijlings, J., Baroni, M., and Sebe, N. (2012). Distributional semantics with eyes: Using image analysis to improve computational representations of word meaning. In *MM'12: Proceedings of the 20th ACM International Conference on Multimedia*, pages 1219–1228, Nara, Japan. ACM. 29, 39

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526. vii, 43, 55, 56

Bullinaria, J. A. and Levy, J. P. (2012). Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907. 38, 46

Burgess, C. (2001). Representing and resolving semantic ambiguity: A contribution from high-dimensional memory modeling. In Gorfein, D. S., editor, *On the Consequences of Meaning Selection: Perspectives on Resolving Lexical Ambiguity*, Decade of Behavior, pages 233–260. American Psychological Association. 44

Bustos, B. and Navarro, G. (2004). Probabilistic proximity searching algorithms based on compact partitions. *Journal of Discrete Algorithms*, 2(1):115 – 134. The 9th International Symposium on String Processing and Information Retrieval. 63

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307. 53, 54, 56

Chandler, D. (2007). *Semiotics: The Basics*. Routledge. 24

Charles, W. G. (2000). Contextual correlates of meaning. *Applied Psycholinguistics*, 21(4):505–524. 25

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–393. 54

Chen, X., Hu, X., Zhou, Z., An, Y., He, T., and Park, E. (2012). Modeling semantic relations between visual attributes and object categories via Dirichlet forest prior. In *CIKM'12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1263–1272, Hawaii, USA. ACM. 39

Cimiano, P., Schultz, A., Sizov, S., Sorg, P., and Staab, S. (2009). Explicit versus latent concept models for cross-language information retrieval. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 1513–1518, California, USA. IJCAI Organization. 49

Clark, S. (2015). Vector space models of lexical meaning. In Lappin, S. and Fox, C., editors, *Handbook of Contemporary Semantics*. Wiley-Blackwell, 2nd edition. 44

Collins, M. and Duffy, N. (2002). Convolution kernels for natural language. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press. 38

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27. 61

Cruse, D. A. (1986). *Lexical Semantics*. Cambridge University Press. 25

Cunningham, P. and Delany, S. J. (2007). *k*-nearest neighbour classifiers. Technical Report UCD-CSI-2007-4, UCD School of Computer Science and Informatics. 62

Curran, J. R. (2004). *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh. 37, 43, 56

Daelemans, W. (1999). Memory-based language processing: Introduction to the special issue. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(3):287–292. 63

Daelemans, W. and van den Bosch, A. (2005). *Memory-Based Language Processing*. Studies in Natural Language Processing. Cambridge University Press. 63

Daelemans, W. and van den Bosch, A. (2010). Memory-based learning. In Clark, A., Fox, C., and Lappin, S., editors, *The Handbook of Computational Linguistics and Natural Language Processing*, pages 154–179. Wiley-Blackwell. 63

Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosc, A. (2009). TiMBL: Tilburg memory-based learner version 6.3 reference guide. Technical Report ILK Technical Report –ILK 10-01, ILK Research Group. 62

De Vine, L. (2013). Some extensions to representation and encoding of structure in models of distributional semantics. Master's thesis, Queensland University of Technology. 52

De Vries, C. M. (2014). *Document clustering algorithms, representations and evaluation for information retrieval*. PhD thesis, Queensland University of Technology. 52

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407. 28, 35, 37, 40, 47

Deza, E. and Deza, M. (2006). *Dictionary of distances*. Elsevier. 54

Deza, M. M. and Deza, E. (2014). *Encyclopedia of Distances*. Springer-Verlag Berlin Heidelberg, 3 edition. 54

Dunbar, R. (1996). *The Trouble with Science*. Harvard University Press. 24

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218. 48

Eddington, D. (2008). Linguistics and the scientific method. *The International Journal of The Linguistic Association Of The Southwest (LASSO)*, 27(2):1–16. 27

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653. 29

Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *EMNLP 2008: 2008 Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference*, pages 897–906, Honolulu, Hawaii. Association for Computational Linguistics. 38

Evgeniou, T., Pontil, M., and Poggio, T. (1999). A unified framework for regularization networks and support vector machines. Technical report, MIT, AI Lab and CBCL, Cambridge, MA, USA. Retrieved from ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1654.pdf. 61

Firth, J. (1957). *Papers in Linguistics 1934–51*. Oxford University Press. 25

Fisher, R. A. (1936). Has mendel's work been rediscovered? *Annals of Science*, 1:115–137. Obtained from https://drmc.library.adelaide.edu.au/dspace/bitstream/2440/15123/1/144.pdf. 27

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI-07)*, pages 1606–1611, Hyderabad, India. AAAI Press. 39, 40

Gallant, S. I. (1991). A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, 3(3):293–309. 42

Gallant, S. I. (1994). Methods for generating or revising context vectors for a plurality of word stems. US Patent 5,325,298. 42

Gallant, S. I. (2000). Context vectors: A step toward a grand unified representation. In Wermter, S. and Sun, R., editors, *Hybrid Neural Systems*, volume 1778 of *Lecture Notes in Computer Science*, pages 204–210. Springer Berlin Heidelberg. 42

Gardner, M., Talukdar, P., Krishnamurthy, J., and Mitchell, T. (2014). Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 397–406, Doha, Qatar. Association for Computational Linguistics. 39

Geva, S. and De Vries, C. M. (2011). TOPSIG: Topology preserving document signatures. In Berendt, B., de Vries, A., Fan, W., Macdonald, C., Ounis, I., and Ruthven, I., editors, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 333–338, Glasgow, Scotland, UK. ACM. 42

Gionis, A., Indyk, P., Motwani, R., et al. (1999). Similarity search in high dimensions via hashing. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529. 52

Gorman, J. and Curran, J. R. (2006). Random indexing using statistical weight functions. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 457–464, Sydney, Australia. Association for Computational Linguistics. 44

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*, volume 278 of *The Springer International Series in Engineering and Computer Science*. Springer US, Norwell, MA, USA, 1 edition. 38

Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato. 47

Hanks, P. (1996). Contextual dependency and lexical sets. *International Journal of Corpus Linguistics*, 1(1):75–98. 25

Harris, Z. S. (1954). Distributional structure. *Word, The Journal of the International Linguistic Association*, 10:146–162. 21, 23, 24, 25

Hartung, M. and Frank, A. (2010). A structured vector space model for hidden attribute meaning in adjective-noun phrases. In *Coling 2010: 23rd International Conference on Computational Linguistics: Proceedings of the Conference*, pages 430–438, Beijing, China. Association for Computational Linguistics/Tsinghua University Press. 38

Hartung, M. and Frank, A. (2011). Exploring supervised LDA models for assigning attributes to adjective-noun phrases. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 540–551, Edinburgh, Scotland, UK. Association for Computational Linguistics. 40

Honkela, T. (1997). *Self-organizing maps in natural language processing*. PhD thesis, Helsinki University of Technology. 51

Huang, W. and Yin, H. (2012). On nonlinear dimensionality reduction for face recognition. *Image and Vision Computing*, 30(4–5):355–366. 51

Irsoy, O. and Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2096–2104. Curran Associates, Inc. 44

Jackson, J. E. (2004). *A User's Guide to Principal Components*, chapter Scaling of Data, pages 63–79. John Wiley and Sons, Inc. 49

Jain, A., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37. 60

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21. 35

Jones, M. N. and Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114(1):1–37. 37

Jones, W. P. and Furnas, G. W. (1987). Pictures of relevance: A geometric analysis of similarity measures. *Journal of the Association for Information Science and Technology*, 38(6):420–442. 57

Jonnalagadda, S., Cohen, T., Wu, S., and Gonzalez, G. (2012). Enhancing clinical concept extraction with distributional semantics. *Journal of Biomedical Informatics*, 45(1):129–140. 38

Jonnalagadda, S., Leaman, R., Cohen, T., and Gonzalez, G. (2010). A distributional semantics approach to simultaneous recognition of multiple classes of named entities. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 6008 of *Lecture Notes in Computer Science*, pages 224–235. Springer Berlin Heidelberg. 40

Jurgens, D., Mohammad, S., Turney, P., and Holyoak, K. (2012). SemEval-2012 Task 2: Measuring degrees of relational similarity. In *First Joint Conference on Lexical and Computational Semantic (*SEM)*, pages 356–364, Montréal, Canada. Association for Computational Linguistics. 36

Jurgens, D. and Stevens, K. (2010). HERMIT: Flexible clustering for the SemEval-2 WSI task. In *SemEval 2010: 5th International Workshop on Semantic Evaluation: Proceedings of the Workshop*, pages 359–362, Uppsala, Sweden. Association for Computational Linguistics. 38

Kamp, H. (2002). A theory of truth and semantic representation. In Portner, P. H. and Partee, B. H., editors, *Formal Semantics: The Essential Readings*. Wiley-Blackwell. 23

Kanejiya, D., Kumar, A., and Prasad, S. (2003). Automatic evaluation of students' answers using syntactically enhanced LSA. In Burstein, J. and Leacock, C., editors, *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 53–60, Edmonton, Canada. Association for Computational Linguistics. 40

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In Gleitman, L. R. and Josh, A. K., editors, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036, Mahwah, New Jersey. Erlbaum. 42, 50

Kilgarriff, A. (2006). Word senses. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*, pages 29–46. Springer Netherlands. 25

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480. 51

Kulis, B. (2013). Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364. 55

Kulkarni, S. and Harman, G. (2011). *An elementary introduction to statistical learning theory*, volume 853 of *Wiley Series in Probability and Statistics*. John Wiley and Sons. 61

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240. 25, 29

Lee, L. (1999). Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 25–32, Maryland, USA. Association for Computational Linguistics. 57

Lee, L. (2001). On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics*, pages 65–72. 56

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *From Context to Meaning: Distributional Models of the Lexicon in Linguistics and Cognitive Science, Special Issue of the Italian Journal of Linguistics*, 20(1):1–31. 24, 25, 26

Leopold, E. (2005). On semantic spaces. *LDV-Forum (Special Issue on Text Mining)*, 20(1):63–86. 48

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Confernce*, pages 64–71, Madrid, Spain. Association for Computational Linguistics. 25

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *COLING-ACL'98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics: Proceedings of the Conference*, volume 1, pages 768–774, Montreal, Quebec, Canada. Association for Computational Linguistics. 40

Lin, D. (1998b). An information-theoretic definition of similarity. In Shavlik, J. W., editor, *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, Wisconsin, USA. Morgan Kaufmann Publishers Inc. 57

Lin, D. and Pantel, P. (2001). DIRT – Discovery of inference rules from text. In *KDD-2001: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328, San Francisco, CA, USA. ACM. 25, 40

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444. 38

Lops, P., de Gemmis, M., Semeraro, G., Musto, C., and Narducci, F. (2013). Content-based and collaborative techniques for tag recommendation: An empirical evaluation. *Journal of Intelligent Information Systems*, 40(1):41–61. 40

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28(2):203–208. 25, 28, 36, 37, 40, 44

Manning, C. D., Raghavan, P., and Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, draft april 1, 2009 edition. 43

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press. 23

Martin, C. D. and Porter, M. A. (2012). The extraordinary SVD. *The American Mathematical Monthly*, 119(10):838–851. 48

Martin, D. I. and Berry, M. W. (2011). Mathematical foundations behind latent semantic analysis. In Landauer, T. K., McNamara, D. S., Dennis, S., and Kintsch, W., editors,

*Handbook of Latent Semantic Analysis*, chapter Mathematical foundations behind latent semantic analysis, pages 35–55. Routledge. 48, 49

Mehdad, Y., Moschitti, A., and Zanzotto, F. M. (2010). Syntactic/semantic structures for textual entailment recognition. In *NAACL HLT 2010: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Proceedings of the Main Conference*, pages 1020–1028, Los Angeles, California. Association for Computational Linguistics. 38

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781. 44

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28. 25

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195. 39

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48. 25

Mostow, J., Chang, K.-M., and Nelson, J. (2011). Toward exploiting EEG input in a reading tutor. In Biswas, G., Bull, S., Kay, J., and Mitrovic, A., editors, *Artificial Intelligence in Education*, volume 6738 of *Lecture Notes in Computer Science*, pages 230–237, Berlin, Heidelberg. Springer Berlin Heidelberg. 39

Murphy, B., Talukdar, P., and Mitchell, T. (2012). Selecting corpus-semantic models for neurolinguistic decoding. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics: Volume 1: Proceedings of the main conference and the shared task*, pages 114–123, Montréal, Canada. Association for Computational Linguistics. 38

Nallapati, R. (2004). Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71, Sheffield, UK. ACM. 30

Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250. 29

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199. 38, 40

Palmer, D. D. (2010). Text pre-processing. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing*, Chapman and Hall/CRC Machine Learning & Pattern Recognition. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2nd edition. ISBN 978-1420085921. 41

Pantel, P. (2005). Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 125–132, Ann Arbor, Michigan. Association for Computational Linguistics. 25

Paradis, C. (2012). Lexical semantics. In Chapelle, C., editor, *The Encyclopedia of Applied Linguistics*. Blackwell Publishing Ltd. 24

Partee, B. H. (2011). Formal semantics: Origins, issues, early impact. In Glanzberg, M., Partee, B. H., and Skilters, J., editors, *Baltic International Yearbook of Cognition, Logic and Communication*, volume 6 of *Formal Semantics and Pragmatics: Discourse, Context, and Models*, pages 1–52. New Prairie Press, Manhattan, KS. 23

Pavoine, S., Vallet, J., Dufour, A.-B., Gachet, S., and Daniel, H. (2009). On the challenge of treating various types of variables: Application for improving the measurement of functional diversity. *Oikos*, 118(3):391–402. 54

Périnet, A. and Hamon, T. (2014a). Distributional context generalisation and normalisation as a mean to reduce data sparsity: Evaluation of medical corpora. In Przepiorkowski, A. and Ogrodniczuk, M., editors, *Advances in Natural Language Processing*, volume 8686 of *Lecture Notes in Computer Science*, pages 128–135. Springer International Publishing. 47

Périnet, A. and Hamon, T. (2014b). *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, chapter Generalising and Normalising Distributional Contexts to Reduce Data Sparsity: Application to Medical Corpora, pages 1–10. Association for Computational Linguistics and Dublin City University. 44, 47

Plank, B. and Moschitti, A. (2013). Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 1498–1507, Sofia, Bulgaria. Association for Computational Linguistics. 38

QasemiZadeh, B. (2015). *Investigating the Use of Distributional Semantic Models for Co-Hyponym Identification in Special Corpora*. PhD thesis, National University of Ireland, Galway. i

QasemiZadeh, B. and Handschuh, S. (2015). Random indexing explained with high probability. In Kral, P. and Matousek, V., editors, *Text, Speech and Dialogue (TSD)*, volume 9302 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 480–489, Pilsen, Czech. Springer International Publishing Switzerland. 50

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *MT Summit IX: Proceedings of the Ninth Machine Translation Summit*, pages 23–27, New Orleans, USA. 28

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 448–453, Montreal, Quebec, Canada. Morgan Kaufmann Publishers Inc. 29

Riordan, B. and Jones, M. N. (2011). Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science*, 3(2):303–345. 30

Roller, S. and Schulte im Walde, S. (2013). A multimodal LDA model integrating textual, cognitive and visual modalities. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1157, Seattle, Washington, USA. Association for Computational Linguistics. 39

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633. 25

Russell, B. (2014). *The Problems of Philosophy*. Bibliotech Press. Originally published by Oxford University Press in 1912. 27

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, USA, 2nd edition. 49

Sahlgren, M. (2005). An introduction to random indexing. Technical report, Swedish ICT (SICS). Retrived from https://www.sics.se/~mange/papers/RI_intro.pdf. 46, 50

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Stockholm University. 24, 25, 29, 30, 36, 37, 46, 48, 50

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54. 26

Sahlgren, M., Holst, A., and Kanerva, P. (2008). Permutations as a means to encode order in word space. In Sloutsky, V., Love, B., and Mcrae, K., editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 1300–1305. Cognitive Science Society, Austin, TX. 37

Sahlgren, M., Karlgren, J., Coster, R., and Jorvinen, T. (2003). SICS at CLEF 2002: Automatic query expansion using random indexing. In Peters, C., Braschler, M., and Gonzalo, J., editors, *Advances in Cross-Language Information Retrieval*, volume 2785 of *Lecture Notes in Computer Science*, pages 311–320. Springer Berlin Heidelberg. 44

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620. v, 34, 35, 40

Schultz, M. and Joachims, T. (2004). Learning a distance metric from relative comparisons. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA. 55

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123. 42

Schütze, H. and Pedersen, J. (1995). Information retrieval based on word senses. In *Proceedings: Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Nevada, USA. 25

Séaghdha, D. O. and Korhonen, A. (2011). Probabilistic models of similarity in syntactic context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1057, Edinburgh, Scotland, UK. Association for Computational Linguistics. 38, 40

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323. 62

Shimizu, N., Hagiwara, M., Ogawa, Y., Toyama, K., and Nakagawa, H. (2008). Metric learning for synonym acquisition. In *Coling 2008: 22nd International Conference on Computational Linguistics: Proceedings of the Conference*, pages 793–800, Manchester, UK. Association for Computational Linguistics. 56

Sichel, H. S. (1975). On a distribution law for word frequencies. *Journal of the American Statistical Association*, 70(351):542–547. 49

Silberer, C., Ferrari, V., and Lapata, M. (2013). Models of semantic representation with visual attributes. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 572–582, Sofia, Bulgaria. Association for Computational Linguistics. 39

Sinclair, J., Jones, S., and Daley, R. (1970/2004). *English Collocation Studies: The OSTI Report*. London: Continuum (Originally mimeo 1970). 25

Singhal, A., Buckley, C., and Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, New York, NY, USA. ACM. 43

Stanford, K. (2013). Underdetermination of scientific theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Winter 2013 edition. 27

Stubbs, M. (2009). Memorial article: John Sinclair (1933–2007): The search for units of meaning: Sinclair on empirical semantics. *Applied Linguistics*, 30(1):115–137. 25

Thater, S., Fürstenau, H., and Pinkal, M. (2010). Contextualizing semantic representations using syntactically enriched vector models. In *ACL 2010: 48th Annual Meeting*

*of the Association for Computational Linguistics: Proceedings of the Conference*, pages 948–957, Uppsala, Sweden. Association for Computational Linguistics. 38

Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346. 44

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188. 24, 36, 39, 42, 43, 46, 48

Turtle, H. R. and Croft, W. B. (1992). A comparison of text retrieval models. *Computer Journal*, 35(3):279–290. 29

Van der Maaten, L. J. P., Postma, E. O., and Van den Herik, H. J. (2009). Dimensionality reduction: A comparative review. Technical Report TiCC TR 2009-005, Tilburg University. 51

Weeds, J., Weir, D., and McCarthy, D. (2004). Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland. Association for Computational Linguistics. 57

Weeds, J., Weir, D., and Reffin, J. (2014). Distributional composition using higher-order dependency vectors. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 11–20, Gothenburg, Sweden. Association for Computational Linguistics. 38

Widdows, D. (2003). Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *HLT-NAACL 2006: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics: Proceedings of the Main Conference*, pages 197–204, New York, USA. Association for Computational Linguistics. 38, 40

Widdows, D. (2004). *Geometry and Meaning*. Number 172 in CSLI Lecture Notes. CSLI Publications, Stanford, CA. 30

Wilks, Y. A. and Tait, J. I. (2005). A retrospective view of synonymy and semantic classification. In Tait, J. I., editor, *Charting a New Course: Natural Language Processing and Information Retrieval*, volume 16 of *The Kluwer International Series on Information Retrieval*, pages 1–11. Springer Netherlands, 1 edition. 35

William J. Gilbert, W. K. N. (2004). *Modern Algebra with Applications*. John Wiley & Sons, Inc., second edition. 33

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2002). Distance metric learning, with application to clustering with side-information. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press. 55

Yamamoto, K. and Asakura, T. (2010). Even unassociated features can improve lexical distributional similarity. In *Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010)*, pages 32–39, Beijing, China. Coling 2010 Organizing Committee. 44

Yang, C. (2013). Who's afraid of George Kingsley Zipf? or: Do children and chimps have language? *Significance: The Magazine of the Royal Statistical Society and the American Statistical Soceity*, 10(6):29–34. 46

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90. 63

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In Fisher, D. H., editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, pages 412–420, Tennessee, USA. Morgan Kaufmann. 46

Zadeh, L. A. (2010). Computing with words and perceptions—A paradigm shift. In Arabnia, H. R., Chiu, S. C., Gravvanis, G. A., Ito, M., Joe, K., Nishikawa, H., and Solo, A. M. G., editors, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010)*, pages 3–5, Nevada, USA. CSREA Press. 29

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics. 39, 44

Zhitomirsky-Geffet, M. and Dagan, I. (2009). Bootstrapping distributional feature vector quality. *Computational Linguistics*, 35(3):435–461. 44