INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
LABORATORIO DE INTELIGENCIA ARTIFICIAL

Automatic elaboration of a search-by-concept dictionary

TESIS

Que para obtener el grado de
Maestro en Ciencias en Ingeniería de Cómputo
con opción en sistemas digitales

Presenta
Ing. Oscar Méndez Martínez

Directores de tesis
Dr. Francisco Hiram Calvo Castro
Dr. Marco Antonio Moreno Armendáriz

Zacatenco, México D.F., julio de 2014

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 20 del mes de mayo de 2014 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"***Automatic elaboration of a search-by-concept dictionary***"

Presentada por el alumno:

Méndez	**Martínez**	**Oscar**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:	A	1	2	0	5	8	6

aspirante de: **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Francisco Hiram Calvo Castro

Dr. Marco Antonio Moreno Armendáriz

Dr. Sergio Suárez Guerra

Dr. Alexander Gelbukh

Dra. Olga Kolesnikova

Dr. Grigori Sidorov

ESTADOS UNIDOS MEXICANOS

PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día **29** del mes de **mayo** del año **2014**, el (la) que suscribe **Oscar Méndez Martínez** alumno(a) del Programa de **Maestría en Ciencias en Ingeniería de Cómputo con Opción en Sistemas Digitales**, con número de registro **A120586**, adscrito(a) al **Centro de Investigación en Computación**, manifiesto(a) que es el (la) autor(a) intelectual del presente trabajo de Tesis bajo la dirección del (de la, de los) **Dr. Francisco Hiram Calvo Castro y Dr. Marco Antonio Moreno Armendáriz,** y cede los derechos del trabajo titulado "**Automatic elaboration of a search-by-concept dictionary"**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del (de la) autor(a) y/o director(es) del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones **oscarq_18@hotmail.com, hiramcalvo@gmail.com y mam.armendariz@gmail.com**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Oscar Méndez Martínez

Nombre y firma del alumno(a)

Resumen

Como parte de la evolución tecnológica que el mundo ha experimentado en los últimos años, el formato electrónico en los diccionarios es una realidad. Su funcionamiento es simple, mapear palabras a sus definiciones. Este enfoque tradicional es de gran ayuda para lectores, pero no toma en cuenta la perspectiva de las personas que producen lenguaje quienes tienden a requerir una búsqueda inversa iniciando con grupos de palabras o términos relacionados. La necesidad de un modo de acceso de búsqueda diferente llevó a la creación del diccionario inverso, el cual realiza un mapeo inverso; es decir, dada una frase que describe un concepto, el diccionario entrega las palabras cuyas definiciones coinciden con la frase de entrada.

Esta tesis presenta un nuevo enfoque para la utilización de un diccionario inverso a través de la creación de un diccionario de búsqueda por concepto basado en la representación de espacios vectoriales utilizando análisis semántico y técnicas estadísticas de procesamiento de lenguaje natural. Las palabras son representadas como vectores numéricos basados en diversas medidas de similitud semántica y medidas probabilísticas, las propiedades semánticas de las palabras son capturadas en los elementos del vector determinados por un contexto lingüístico. Tres fuentes para la creación de vectores de las palabras fueron utilizadas: WordNet, un tesauro distribucional y el algoritmo de distribución latente de Dirichlet; cada fuente constituye un espacio semántico. Las entradas al diccionario consisten en conceptos de n-sustantivos. Cada sustantivo es sustituido por su vector numérico para posteriormente llevar a cabo un análisis del espacio semántico con el fin de desplegar una lista de palabras como salida del diccionario. Se creó un conjunto de prueba con 50 conceptos para evaluar el desempeño del sistema. Comparando los resultados experimentales del diccionario contra los provistos por OneLook Reverse Dictionary se demostró que el primero ofrece mejores resultados que implementaciones actualmente disponibles.

Abstract

As part of the technological evolution the world has experienced during the last years, dictionaries are now available in electronic format. Its performance is simple, just mapping words to their definitions. This traditional approach is really helpful mostly for readers and language students, but is not good enough taking into account the perspective of people who produce language. A language producer tends to require a reverse search that starts with a group of words or a series of related terms, looking for a target word. The need for a different search access mode led to the creation of the reverse dictionary which performs an inverse mapping; i.e., given a phrase describing a desired concept, it provides words whose definitions match the entered definition phrase.

This thesis presents a new approach for reverse dictionary creation through the development of a search-by-concept dictionary based on vector space representation using semantic analysis and statistical natural language processing techniques. Words are represented as numeric vectors based on different semantic similarity measures and probabilistic measures; the semantic properties of a word were captured in the vector elements determined by a given linguistic context. Three sources were used for word vector construction: WordNet, a distributional thesaurus and the Latent Dirichlet Allocation algorithm; each source constituted a Semantic Space.

The search-by-concept dictionary input is conformed by a concept of n-nouns. All input members are read and substituted by their corresponding vectors. Then, a semantic space analysis including a filtering and ranking process is carried out to display the dictionary output conformed of a list of target words. A test set of 50 concepts was created in order to evaluate system's performance. Comparing the experimental results against OneLook Reverse Dictionary demonstrates that the search-by-concept dictionary provides better results over current available implementations.

Acknowledgements:

A very special thanks to my mother María Leticia Méndez Martínez. Words cannot express how grateful I am for all the sacrifices you have made on my behalf. You have always been there for me and supported me whenever I needed.

I would like to thank my family for your unconditional support, you encouraged me to overcome the difficulties experienced in these two years of studies and research.

I would like to express my special appreciation and thanks to my advisors Dr. Hiram Calvo Castro and Dr. Marco Antonio Moreno Armendáriz for all your support given during this investigation and the valuable knowledge transmitted which is reflected in this research. Your advice have been priceless and allowed me to grow as a person and as a professional.

I would also like to thank my laboratory friends for your support and all the good moments shared during my stay, especially Josafath Israel Espinosa Ramos and Alfonso Martínez Cruz who were always willing to help and give me their best suggestions. I will always appreciate the valuable time that both spent helping me during my writing process.

A very special thanks to my girlfriend Andrea Sanchez Pulido for all your love and support during my thesis writing and the last part of my research. You have been a constant source of motivation and encouragement. I love you.

Finally, I would like to thank to Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC) and CONACYT for providing me the necessary support to carry out this research work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Historically, human intelligence involving thought processes, reasoning and behavior has been studied by scientists in order to build intelligent entities. As a consequence, artificial intelligence (AI) was born as a new field in science and engineering. There is no unique definition for this subject, different approaches exist evaluating mainly two aspects: human behavior and rationality. The first one seeks to create a machine able to think (make decisions, solve problems, learn, etc.) and act like humans; the second one relies in mathematics and engineering to achieve computer intelligence capable of an ideal thinking and behavior.

The AI field has a wide variety of research topics such as: knowledge representation, automated reasoning, natural language processing, machine learning, computer vision, robotics, among others. This research is focused in natural language processing (NLP), an hybrid field originated from the joint work of modern linguistics and AI researchers; it is also known as computational linguistics. Understanding language requires an understanding of the subject matter and context, not just an understanding of the structure of sentences [39]. This is not an easy task and implies work within different subjects such as language models creation, information extraction, text classification, information retrieval, just to mention a few. The size of the application areas within NLP is proportional to the complexity of the language understanding problem. The main area related with this project is information retrieval (IR).

Information retrieval is a subfield of NLP concerned about finding documents that are relevant to a user, based on his information needs, through the usage of algorithms and models. Then, having a natural language query specifying the user's information needs, an IR system needs to map it to a set of documents which meet the user needs, and deliver them sorted by relevance. There are two main models to do this, one based on exact match between the query and output information; the second one which ranks documents according to their estimated relevance to the query. There are many examples of IR systems, being the most common the search engines for the Internet. However, there is an originally printed resource whose electronic version has been improved due to the implementation of information retrieval techniques. This resource is the traditional dictionary.

Over the years, people have used dictionaries for two well-defined purposes. Both of them are reflected on the dictionary's definition [29]:

"A collection of words listed alphabetically in a specific language, which contains their usage information, definitions, etymologies, phonetics, pronunciations, and other linguistic features; additionally it could be formed of a collection of words in one language with their equivalents in another, also known as a lexicon."

With these different ideas coming together, this resource has not lost importance and continues to be widely used around the world.

As part of the technological evolution the world has experienced during the last years, dictionaries are now available in electronic format. This resource has different advantages over the traditional printed dictionary, being the most important the easy access allowed to users and the very fast response time. Lexicographers constantly improve this resource, in order to assist language users, by increasing the number of words defined in the dictionary and adding more information associated with each one of them. The dictionary performance is simple, just mapping words to their definitions, *i.e.* it does a lookup based on the correct spelling of the input word to find the definition.

This traditional approach is really helpful mostly for readers and language students, but is not good enough taking into account the perspective of people who produce language. We all have experienced the problem of being unable to express a word that represents an idea in our mind although we are conscious of related terms, a partial description, even the definition. This may be due to a lack of knowledge in the word's meaning or a recall problem. People mainly affected by this problem are writers, speakers, students, scientists, advertising professionals, among others. For them, traditional dictionary searches are often unsuccessful because these kind of search demands an exact input, while a language producer tends to require a reverse search where the input are a group of words forming a formal definition or just a series of related terms, and the output is a target word.

The need for a different search access mode in a dictionary led to the creation of the reverse dictionary. Its basic objective is to retrieve a target word when a group of words which appear in its definition are entered. In other words, given a phrase describing a desired concept or idea, the reverse dictionary provides words whose definitions match the entered phrase. The chances of giving an exact definition of a concept are very difficult so synonym words or related words could also be considered during the search.

Existing reverse dictionaries general performance is based in classical IR techniques. Unfortunately, many of the well-known syntactic search problems (polysemy, synonymy, complex concepts) are part of their operation and the quality of their results tend to be disappointing. In order to outcome this difficulties, this research develops a new method to generate a reverse dictionary based on a large lexical English database known as WordNet and the implementation of different semantic similarity measures which help us in the generation of a semantic space.

1.1 Motivation

People have interacted with dictionaries for a long time and the availability of their electronic versions has benefited new generations because of the advantages they offer over the traditional printed format and the possibility for new search access modes. Reverse lookup became a reality for electronic dictionaries implementing IR techniques but their outputs are not accurate enough.

On the other hand, vector-space word representation has demonstrated a good performance capturing syntactic and semantic regularities in language. Its advantage over other models is the level of generalization achieved due to its distributed representation; this is not possible with classical n-gram language models [30].

Those are the principle motivations for this thesis; to improve the performance of an existing NLP application with a new approach of reverse search inspired on the advantages that vector-space word representation has demonstrated when applied on different tasks.

1.2 Problem to solve

The structure of a reverse dictionary is, as its name says, the opposite of the traditional form. Here the input is a definition or a part of it, even a group of words conceptually related; while the output consists of a list of words whose definitions correspond to the description given at first, either through an exact match or assuming similarity based on a semantic analysis. Hence, the purpose of a reverse dictionary is the mapping of definitions to their words.

The approaches given to this NLP application research converge to the same root, the usage of information retrieval techniques. Specifically, the reverse lookup of the dictionaries has been based on syntactic search, *i.e.* using words or phrases as a query, the search procedure consists basically in syntactically matching the input definition with a target word definition. In some cases, a semantic analysis is done but only after the syntactic analysis process.

Several problems have been noticed to affect the performance of the reverse search using classical IR techniques, as listed below.

Polysemy. This occurs when a word has different senses, therefore, a reverse dictionary output may contain target words in whose definitions the query word is used in a different sense from what the user refers.

Synonymy. Different words can express the same meaning when they are used in a specific context, in such cases they are synonyms. Using exact matching during reverse search does not allow the presence of target words whose definition could contain synonyms of the terms used in the input.

Complex concepts. Natural language phrases may be used to form complex concepts which need to be analyzed as a phrase to understand their meaning. Syntactic analysis evaluates a query as a list of words, discriminating them and losing the meaning of the complex concept; this affects negatively the quality of the output.

1.3 Justification

Traditional dictionaries have benefited humanity for years, but this benefit is mostly appreciated from the reader's point of view. In this scenario the lookup is based in single words in order to get their corresponding meanings; this is not always helpful when viewed from the language producer's perspective. In this case, people (speakers, writers, etc.) needs are the opposite; initially their minds are full of meanings or concepts and their goal is to find a word able to represent their thinking. A reverse dictionary allows this kind of entry structure and the research for this application has been growing with the availability of dictionaries in electronic format, but their performance has several flaws due to the presence of problems during syntactic search.

The implementation of vector-space word representations enables the removal of most well-known syntactic problems through their capacity to capture syntactic and semantic regularities in language [30]. Also, working with continuous space models permits distributed representation with good levels of generalization and includes the characteristic that similar words tend to have similar vectors.

1.4 Hypothesis

Having a concept formed of n nouns as input to the reverse dictionary, each noun is represented as a numeric vector in order to locate them in a vector space of n dimensions determined by the elements included on the vector; this vector space was previously conformed by the vector representation of a huge vocabulary. Given the input set of points represented in the n-dimensional vector space, vector algebra is applied to find a new point from which a sample of the nearest neighbors is taken. The words included in this sample should mix semantically the characteristics described by the original entries, representing the target words from the reverse lookup process of the search-by-concept dictionary.

1.5 Objectives

1.5.1 General objective

To develop a method that automatically generates a search-by-concept English dictionary from several sources of semantic relatedness measures.

1.5.2 Particular objectives

1. To generate a numeric representation of words by means of a semantic vector space.

2. To create a semantic space based on:
   (a) WordNet
   (b) Distributional Thesaurus
   (c) Latent Dirichlet Allocation
3. To determine which of these models is the closest to human associative reasoning.

1.6 Proposal

This thesis presents a new approach for reverse dictionary creation, one based on vector space representation using semantic analysis and statistical NLP techniques. Words are represented as numeric vectors based on different semantic similarity measures and probabilistic measures, where the elements of the vectors consists in the similarity between each word of the vocabulary and a set of topics previously defined. The space containing all word vectors is called semantic space.

The semantic space is created from three different sources, in order to get an analysis from different approaches: supervised approach assisted by WordNet, semi-supervised approach assisted by a distributional thesaurus and unsupervised approach assisted by the Latent Dirichlet Allocation (LDA) algorithm. This results in the creation of three databases, one for each source.

The search-by-concept dictionary input is conformed by a concept of n-nouns, this concept should represent an idea or a definition, but it could also be a random list of words. All input members are read and substituted by their corresponding vectors. Then, the average vector is calculated and used to get a sample of the nearest neighbors. The sample of word vectors passes through a filtering and ranking process, ending with a dictionary output conformed by a list of n target words defined by the user. This proposal is summarized in Figure 1.1 through its flowchart.

Once the search-by-concept dictionary is ready, a test set is created to prove its efficacy and an evaluation procedure will determine which one of the models used for the semantic space creation is the closest to human associative reasoning.

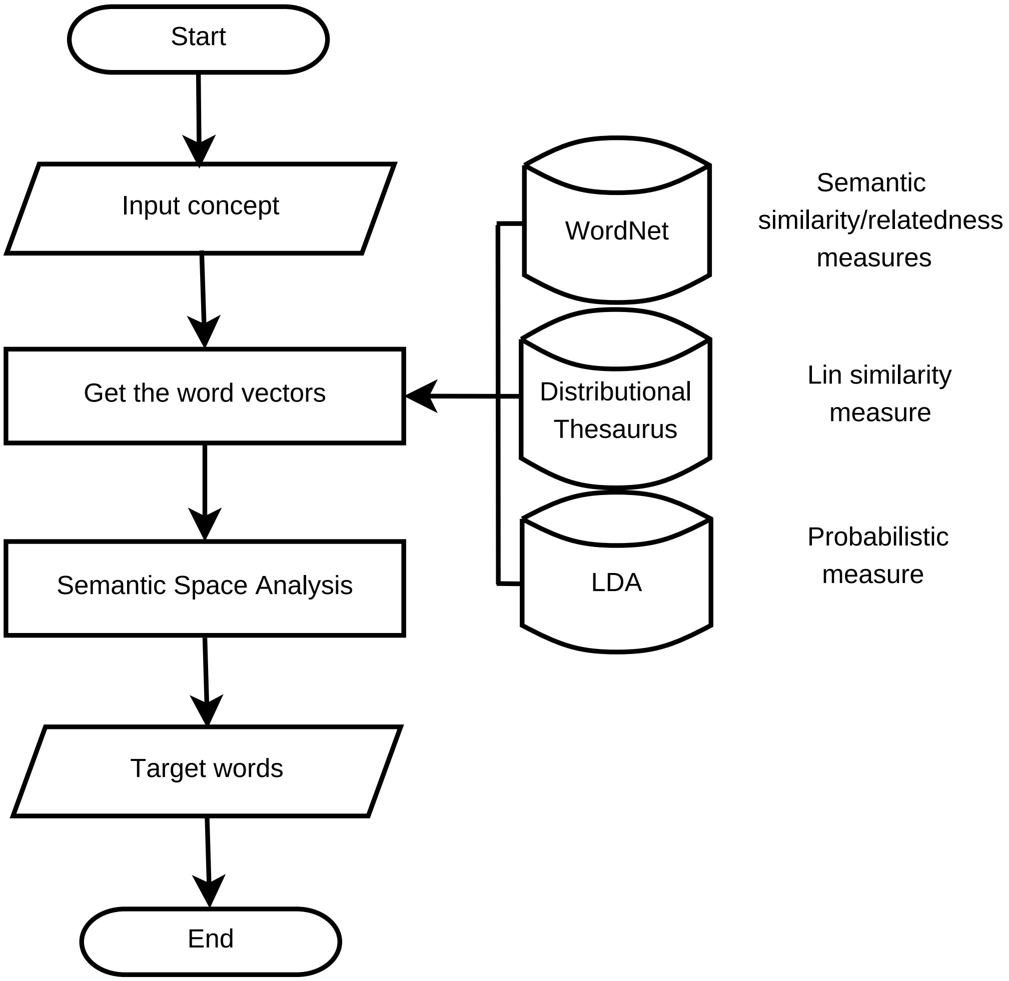


Figure 1.1: Proposal flowchart

1.7 Contributions

1. A search-by-concept dictionary system based on vector space representation using semantic analysis and statistical NLP techniques

2. Three databases of vector-space word representation based on different models:

   (a) WordNet

   (b) Distributional Thesaurus

   (c) LDA

3. The Wikipedia corpus processed for LDA environment.

Chapter 2

State of the Art

Only three printed reverse dictionaries exist for English language [23][16][4]. The reason is probably the complexity of its elaboration, especially the fact of choosing the proper form to distribute the information. The Bernsteirn's Reverse Dictionary [4] was the first of its kind; in this book, the definitions of 13,390 words were reduced to their most brief form and then ordered alphabetically. In order to cover all routes to find a target word, some words have multiple references as the author re-orders the word sequence of the definitions in every possible form. However, the briefness in the definitions could be seen as a limitation as this forced reduction may lead to information loss; although necessary because longer definitions might difficult the compiling work task.

With the availability of dictionaries in electronic format, the interest for a reverse lookup application has been growing during the last years. Unlike printed versions, complications related to information order, concept hierarchy or entries structure disappeared and several attempts have been made in the creation of the reverse lookup method seeking for the best performance.

The first attempts on reverse dictionary creation were based on Boolean operators, *i.e.*, exact match systems. They received a definition or a list of words to begin the reverse search, target words containing the exact form of the input were displayed as output; however, this scenario had few possibilities of occurrence evidencing their limited performances. The Merriam Webster's Collegiate Dictionary in its first electronic versions included a reverse search option based on this procedure.

Another approach for reverse search was done in 1995 with the United States patent of Crawford et al. titled "Reverse electronic dictionary using synonyms to expand search capabilities” [12]. In this work, synonyms were used to expand search capabilities. The dictionary operates in two phases:

- Phase I – numeric codes establishment and assignment.
- Phase II – reverse search sequence.

In Phase I, they create a dictionary database (DB) with words numerically encoded for quick and easy access assisted by the Webster’s Collegiate Thesaurus and Webster’s Collegiate Dictionary (both on electronic version); adding also synonym groups in order to extend the searching process. The thesaurus was needed to establish the synonym group numeric codes and the dictionary to codify all the main entry words along with their definitions. Once the numeric codes are defined, the dictionary database is complete and the word retrieval system is ready to operate.

In Phase II, for every search the numeric codes of the input words are found and stored. However, the input can not be longer than a combination of two words, an enormous limitation. Then, main entry words having numeric codes of the input words within their definitions are located and displayed as output candidates. Finally, after all candidate words are displayed, the process ends; if no candidate words are displayed, the user is instructed to introduce a different combination of words.

The magnitude of this natural language application is appreciated when dictionaries for different languages are constructed (Bilac et al., 2004) [5]. This Japanese reverse dictionary considers as basic principle the comparison between the input phrase and the definitions from a concept dictionary. Before doing the lookup process, they parsed all dictionary definitions with a morphological analyzer in order to generate frequency files which reflect the term frequencies in each definition.

The operation of their dictionary is based on traditional IR metrics. They consider the user input as a query q and the dictionary definitions as the documents d, both composed of terms t. Each user input is represented as a vector $\vec{wq}$ whose elements consist of term frequencies (tf) values and each dictionary definition is represented as a vector $\vec{wd}$ whose elements consist of the product of term frequency and the inverse document frequency (tf·idf). Having data in this form, it is possible to measure similarity between them. In this case, three standard similarity metrics of IR were used:

1. The dot product of vectors, normalized by the sum of term frequencies in the document. The formula is expressed in Equation 2.1.

$$sim(q,d) = \frac{\vec{wq} \cdot \vec{wd}}{\sum_{n=1}^{m} tf(t_n, d)} \tag{2.1}$$

2. The cosine similarity measure. In this formula the dot product is normalized by the product of vector lengths as shown in Equation 2.2.

$$sim(q,d) = \frac{\vec{wq} \cdot \vec{wd}}{|\vec{wq}| \cdot |\vec{wd}|} \tag{2.2}$$

3. The modified cosine measure. For this measure, the vector elements values are replaced by binary values as given in Equation 2.3. Then, the resulting vector is used to calculate the similarity using cosine similarity measure.

$$\begin{gathered} a(t,q) = \begin{cases} 1 \text{ term } t \text{ is in the query} \\ 0 \text{ otherwise} \end{cases} \\ \vec{wq} = (a(t_1,q), ..., a(t_m,q)) \end{gathered} \tag{2.3}$$

The output consists of those words whose definitions have the highest similarity with the user input. All of these measures were used as part of the reverse search process but each one evaluated separately. There is also an attempt to expand the dictionary definitions of a concept by adding the definitions of its hypernyms; this is possible due to the characteristics of the concept dictionary used. And, following their basic principle, a direct checking for a match within the user input and the concept definition is considered as another option of their dictionary output.

A different reverse lookup method was created in [15] (Dutoit et al., 2002). In this proposal, a lexical database of French words called 'The Integral Dictionary' (TID) acts as the main source for the reverse search operation. TID is a semantic network associated to a lexicon with a size comparable to WordNet [32], one of the most important lexical databases for English.

The Integral Dictionary organizes words into a diversity of concepts, classified into categories being only used by this reverse search algorithm: the classes and the themes. Classes form a hierarchy and are annotated with their part-of-speech and themes are concepts that can predicate the classes. As a graph of concepts, ontological concepts are the basic components of TID and each concept is annotated by a gloss of few words describing its content.

TID also includes the implementation of different semantic lexical functions which allows the generation of word senses from another word sense given as an input. Furthermore, TID adds the utility of componential semantics which corresponds to the decomposition of the words into a set of smaller units of meaning: the semes. The structure as a hierarchical graph of concepts, the use of lexical functions combined with a componential semantics analysis gives TID the usefulness for this NLP task.

The algorithm does a reverse search using two main mechanisms.

1. Extraction of sets of words from the database that delimit the search space.
2. Computation of a semantic distance between each word in the delimited search space and the input definition.

In the first mechanism the word sets are extracted using a function that finds for a given word all the hyponyms of one of its hypernyms, this led to an important reduction of the search space. Specifically, in TID it corresponds to the *ToClassSpecific* and *ToClassGeneric* functions. For example, in the definition 'a person who sells food' all the sets of persons are extracted.

The second mechanism implies a more elaborated environment; it views TID as a semantic network. TID superimposes two graphs. The first graph forms an acyclic graph with words as terminal nodes and concepts representing the other nodes. The second graph connects the words using lexical functions. In this algorithm the distance between two words or phrases is derived from the first graph and consists in the sum of two values called by them the semantic activation and the semantic proximity.

Semantic activation of two words, M and N is defined by their set of least common ancestors LCA in the graph, being the semantic activation paths the paths linking both words M and N through each node in the set of least common ancestors. Finally, they define the semantic activation distance as the number of arcs in the activation paths divided by the number of paths. It is important to mention that the LCA allows the delimitation of small concept sets.

On the other hand, the semantic proximity between two words, *M* and *N*, uses sets of asymmetric ancestors which they named the Least Asymmetric Ancestors (LAA). *LAA(M, N)* is the set of nodes that are common ancestors of both words, that are not member of the LCA set and where each member of the LAA set has at least one child, which is an ancestor of *M* and not an ancestor of *N*. However, the *LAA(M,N)* are different from *LAA(N,M)* most of the time, being necessary both calculations for the semantic proximity distance. They also define the semantic asymmetry as the sum of distances of *M* to all the members of both LAA sets and *N* to all the members too. Finally, they define the semantic proximity as the sum of the semantic activation and the semantic asymmetry.

Once both mechanisms were executed, they displayed target words based on their proximity with the input definition. They used these values to rank their output, lower numbers indicate better relevance.

Another proposal for reverse search tries to emulate the behavior of human mind [50] assuming that knowing a word does not imply that a person is able to access it in time, regardless of having it stored in memory (Zock et al., 2004). People use various methods to start a search process in their mind, it could be words, concepts, partial descriptions, related terms, etc. Based on the notion of association which considers that every idea, concept or word is connected; people should have a highly connected conceptual-lexical network in their mind. As a result, any word or concept has the potential to evoke each other.

The goal of this proposal is to build an associative network by enhancing an existing electronic dictionary with syntagmatic associations obtained from a corpus, representing the information of 24 months contained in a mayor French newspaper '*Le Monde*' with a size around 39 million words. First, an extraction of lexical co-occurrences is done to build a network. This network is used by a topic analyzer which performs three tasks: text segmentation into topically homogeneous segments, selection in each segment of the most representative words of its topic and creation of a set of words from the co-occurrence network to expand the selected words of the segment.

Once the topic analyzer process finished, a set of segments and a set of expansion words for each one of them is obtained. The association of the selected words of a segment and its expansion words is called a Topical Unit. The Topical Units passed through a double filtering, the first one to discard heterogeneous Topical Units, and the second one to keep the expansion words that satisfy a determined co-occurrence threshold.

After the filtering process, each Topical Unit gathers a set of words supposed to be strongly coherent from the topical point of view. The co-occurrences between these words for all Topical Units filtered are taken, concluding with a new network of topical co-occurrences. This network is proposed to enrich an existing dictionary such as WordNet by adding certain links, in particular on the syntagmatic axis. Considering these links as associations, they will help finding concepts or words related to a given input word [17].

Although this research considers lexical access based on phrases or definitions as part of the human mind behavior, at the end, the target word is accessed via a source word. The mental lexicon is viewed as a huge semantic network composed of nodes (words) and links (associations), with either being able to activate the other; therefore, achieving a target word involved the entrance to the network following the links leading from the input word (source node) to the target word. The contribution is a new index based on the notion of association which needs to be added to an existing electronic dictionary, this satisfies the need to get an adequate mean to reveal the stored information in electronic dictionaries in order to support language producers (speakers, writers, etc.) to find the word they are looking for.

After a few years an improved system for reverse search (Zock et al., 2008) was detailed in [51]. Again, the main concern was finding a correct manner to index the dictionary in order to gain a quick and intuitive access to words. This system allows lexical access based on underspecified input through the creation of a corpus-based association matrix, composed of target words and access keys. In detail, the association matrix consisted of a lexical matrix with one axis containing all the words of the language representing the target words t_w, and the other axis containing the access words a_w representing the words or concepts capable and likely to evoke the target words.

An interesting difference from traditional co-occurrence matrix is that instead of putting a Boolean value at the intersection of the t_w and the a_w, the intersections stores weights and type of links holding between co-occurring terms. Also, they propose the use of lexical functions in order to reduce the number of possible candidate words (output), as a function of the underspecified input, *i.e.*, the number of words given by the user. Mel'čuk adopted the term lexical functions to refer to the fact that two terms are systematically related [48]. As a result, lexical functions encode the combinability of words. Different categories of lexical functions are shown below:

- Paradigmatic associations: hypernymy, hyponymy, synonymy, antonymy, etc.
- Syntagmatic associations: collocations.
- Morphological relations: terms being derived from another part of speech.
- Sound-related items: homophones, rhymes.

The lexical functions handle all of them. However, the experiments only make use of the neighborhood function which produces a set of co-occurring terms within a given window. The usage of lexical functions is justified by the following points:

1. The user is able to specify the type of relation wanted.
2. The list of target words is reduced with a larger number of input words, as applying lexical functions to the input words considering just the intersection of the obtained sets to be relevant target words.

Two different sources were selected as corpus for the association matrix construction: WordNet and Wikipedia. Some advantages of using WordNet as a corpus were mentioned: no need to identify sentence boundaries, avoid semantic ambiguity due to the fact that words are tagged and no need for lemmatization. However, there were problems with the size of vocabulary working only with 63,941 words, losing syntagmatic associations encoding encyclopedic knowledge.

On the other hand, the properties of Wikipedia used as a corpus demonstrated to be the exactly opposite of WordNet. While it contains syntagmatic associations due to its encyclopedic knowledge, it is purely raw text. As a result, text segmentation, lemmatization and the use of stopwords were needed.

Having the corpus, the neighborhood function was applied and the co-occurrences were stored in a database, together with the weight (number of times the two terms appear together) and the type of link. These latter two members were used for output ranking. At the end, given a set of input words, the system provides the user a list of words co-occurring with the input terms.

The most recent reverse dictionary system found in [43] is called the Wordster Reverse Dictionary (Shaw et al., 2013) and was built taking into account two constraints:

1. The user input is unlikely to exactly match the definition of a dictionary word.

2. The response time of an input query needs to be minimum in order to create an application capable of supporting online interaction.

For them, the main challenge consisted in solving a concept similarity problem but one with different characteristics from the concept similarity work reported in the literature where the similarity of concepts model concepts as single words. For a reverse dictionary, a similarity between multiword phrases is necessary. The system operation was divided in two phases, first a selection of candidate words is carried out from a forward dictionary data source (WordNet), and then the candidate words are ranked in order of quality of match.

As mentioned above in this section, the basic approach to solve the reverse search problem consists in a comparison between the user input phrase and all the definitions of a dictionary. In this work, an improvement is done by reducing the set of definitions needed for comparison with the user input phrase. To achieve this, an index that mapped from a word to all the dictionary words in whose definitions it appears was created and used to limit the set of definitions needed for comparison. The index was called a reverse mapping and ensures to keep only the dictionary words in whose definitions the input word being analyzed is contained. Before the reverse mapping, a stemming process is done assisted by the Porter stemmer [35] (a standard stemming algorithm) which reduces each word to its base form by removing common modifications for subject-verb agreement, or variation in parts of speech.

With the reverse mapping sets created for every word in the dictionary, the reverse search is able to initiate. Having an input phrase, stop words are removed from it, then instances of negation words are identified; if the phrase contains negation instances, the query is expanded including antonyms of the negation term. Finally candidate output words are obtained analyzing the reverse mapping sets of the remaining input words.

If a sufficient number of output words (defined by an input parameter) is not generated, the lookup scope is expanded in order to consider another types of conceptually related terms: synonyms, hypernyms and hyponyms. If the number of output words is not reached yet, terms are removed from the query set one by one, starting with those terms appearing in most definitions of the dictionary words. This removes the most common words first, based on the assumption that the least commonly occurring words will be the most important in finding quality output words.

When the number of output words is achieved, a ranking process is done as the last step of the system operation. Words are sorted in order of decreasing similarity to the input phrase based on their semantic similarity. The semantic similarity considers two aspects.

The first one measures the similarity between two terms based on their locations in the WordNet hierarchy. In this case, two terms have little similarity if their least common ancestor LCA in WordNet hierarchy is the root and greater similarity the deeper their LCA is in the hierarchy. The formula is shown in Equation 2.4.

$$\rho(a,b) = \frac{2 * E(A(a,b))}{E(a) + E(b)} \tag{2.4}$$

where the terms are represented by a and b. The function *A(a,b)* returns the least common ancestor shared by both a and b in the WordNet hierarchy and *E(t)* returns the depth of a term t in the WordNet hierarchy. The similarity *ρ(a,b)* is based on [49].

The second aspect for semantic similarity consisted of a similarity measure that is weighted by the importance of each term in the context of the phrase. To generate the importance of each term, a parser (OpenNLP) was used in order to get the grammatical structure of the sentence. Words in the input phrase appearing higher in the parse tree are more important than those appearing at the bottom.

Once both aspects of similarity were measured, a weighted similarity factor takes into account the product of both values. The weighted similarity factor is generated for each term pair *(a,b)* where a belongs to the user input and b belongs to the candidate word definition. All term pair values are used to create a weighted similarity matrix used as input to a generic string similarity algorithm described in [43] to obtain a phrase similarity measure. This value is used to rank candidate words, returning as system output the best matches.

Regarding the investigation carried out in Mexico about reverse dictionaries, there is a project being developed in [44] (Sierra et al., 2011) called *DEBO* which implies the creation of an electronic dictionary for onomasiological search. The concept of onomasiological search refers is equivalent to the concept of reverse search managed in this investigation and in the state of the art. Being a project still in construction, it is not possible to describe their reverse search algorithm; however, their proposal parts from the same root as most of the works described before where the definitions obtained from different sources (dictionaries, encyclopedias, etc.) are used for matching the users' queries. Then, the success or failure during the reverse search totally depends on the variety and accuracy of the definitions stored in their lexical knowledge base.

These were the most interesting proposals found during investigation. Analyzing the algorithms of reverse search, it is visible that a common methodological baseline has been followed, i.e., there has been a tendency during reverse lookup algorithms creation until now. However, all of them included new relevant features in order to improve the existing state of the art performance.

One of the common aspects is the usage of an electronic dictionary to build their databases and the capacity of query expansion including as part of the input phrase different terms conceptually related (synonyms, antonyms, hypernyms and hyponyms) [12] [5] [15] [51] [43]. However, the reverse search done by [12] [5] [51] and [43] at some point of their procedure perform a comparison between the user input phrase and the definition of dictionary target words looking for exact matching, while [15] based its reverse search on the highest similarity values measuring graph distances and [17] based its reverse search following association links from a source word to a target word.

A detailed description of the methodological features contained on the systems mentioned above is shown in Table 2.1, adding a column for the reverse dictionary system proposed in this work in order to show an initial review of some of its characteristics.

As shown in Table 2.1, a lot of progress have been accomplished over the last years following a common methodology, yet more can be done and new approaches may be proposed in order to improve existing performance and results.

Table 2.1: State of the art approaches (+ means other authors, SCD is our proposal)

Feature \ System	Crawford+ 1997	Bilac+ 2004	Dutoit+ 2002	Zock+ 2004	Zock+ 2008	Shaw+ 2013	SCD
Usage of query expansion	X	X	X	X	X	X	
Input phrase of n-terms		X	X				X
Dictionary for DB creation	X	X	X		X	X	X
Thesaurus for DB creation	X						X
Digital encyclopedia for DB creation					X		X
Digital newspaper for DB creation				X			
Vector representation		X					X
Implementation of IR metrics		X		X	X		
Comparison of input with dictionary definition	X	X			X	X	
Search space delimitation			X			X	X
Usage of semantic similarity functions			X			X	X
Usage of a hierarchical graph			X		X	X	X
Implementation of topic segmentation of data				X			X

Chapter 3

Fundamentals

In this chapter, the knowledge necessary to understand the content of this work is described through the definition of basic concepts and full explanations of the different resources used to develop the proposed system; also, a description of the model being used to represent the system data is included giving its formal definition and mentioning how its relation with different hypotheses allows an effective way to represent semantics.

3.1 Vector Space Model

Computers understand very little of the meaning of human language. Recent progress in technology studies the surface of human language, however, a need for deeper semantic technologies is emerging and the vector space model (VSM) is part of the new semantic technologies. In this work, the term 'semantic' is used in a general sense, as the meaning of any word, phrase, sentence or text pertaining to human language.

Originally developed for information retrieval systems [40], VSMs have demonstrated to be useful representing lexical meaning in different natural language processing NLP tasks, such as automatic thesaurus extraction [24], text segmentation [11], word sense discrimination [42], and also have demonstrated to perform well on tasks that involve measuring similarity of meaning between words, phrases, and documents [28].

Researchers involved in the study of semantics have reached the conclusion that the meaning of words is closely connected to the statistics of word usage [18]. The success of VSMs lies in their ability to represent word meaning using distributional statistics. The semantic properties of words are captured in a multi-dimensional space by vectors that are constructed from a given linguistic context. The goal of the VSMs is to represent an object as a point in a space (a vector in a vector space). In this space, points that are close together are semantically similar and points that are far away are semantically distant. So, the semantic similarity between any two points can be calculated directly using a distance measure such as cosine, Euclidean distance or other user-defined measures.

The VSMs have a close relation with the distributional hypothesis which says that words occurring in similar contexts tend to have similar meanings [18]. The intention of applying this hypothesis to concrete algorithms about measuring similarity of meaning had led to VSMs where the interpretation would be that words having similar vectors would tend to have similar meaning.

There are various forms of VSMs and they are subsumed by a general hypothesis called the statistical semantics hypothesis which says that statistical patterns of human word usage can be used to figure out what people mean [20]. This general hypothesis underlies different more specific hypotheses including the distributional hypothesis mention above, a fundamental basis of this work.

Two forms of VSMs will be explained in this work; first the original implementation given to VSMs which performs well in information retrieval and is focused in documents similarity. Then, a variation focused on word similarity measurement, this last form is the one implemented in this work.

3.1.1 The Term-Document Vector Space Model

Having a large collection of documents, it could be organized into a matrix with rows representing terms (usually words) and columns representing documents. This kind of matrix is called a term-document matrix.

In a term-document matrix, the document vectors are represented by the columns in a bag of words form (text represented as a set of words disregarding its grammar and word order) as the row terms have not a specific order, generally after removing the stopwords and a lemmatization process. So, having a set of bags represented in a matrix M, each column $m_{:j}$ corresponds to a bag, each row $m_{i:}$ corresponds to a unique word, and the element m_{ij} corresponds to the frequency of the i-th word in the j-th bag. This is justified considering the bag of words hypothesis [47] which says that the relevance of documents to a query could be estimated by representing the documents and the query as bags of words. In other words, the frequencies of words in a document tend to indicate the relevance of the document to a query. It is important to mention that using frequencies is the simplest form to fill the term-document matrix, the value of element m_{ij} may be carried out by another function such as tf·idf, pointwise mutual information PMI, among others.

To illustrate this type of VSM, a term-document matrix consisting of four documents, being each document a sentence, is shown in Table 3.1.

```
d1 - The boy eats gum.
d2 - The girl dances in the house with another girl.
d3 - The boy jumps the wall.
d4 - The girl eats fish in the house.
```

Table 3.1: Term-document matrix

Terms \ Documents	d1	d2	d3	d4
boy	1	0	1	0
eat	1	0	0	1
gum	1	0	0	0
girl	0	2	0	1
dance	0	1	0	0
house	0	1	0	1
jump	0	0	1	0
wall	0	0	1	0
fish	0	0	0	1

In spite of the crude representation of documents, vectors seem to capture an important aspect of semantics through frequencies as shown in Table 3.1. A possible justification for the term-document matrix may be that the topic of a document will probabilistically influence the author's choice of words when writing the document [47]. Recent generative models, such as Latent Dirichlet Allocation LDA (see section 3.6), directly model this intuition. If two documents have similar topics, then the two corresponding column vectors will tend to have similar patterns of numbers.

3.1.2 The Word-Context Vector Space Model

Instead of measuring document similarity, focusing on word similarity measurement is possible by a simple shift in the matrix interpretation looking at row vectors instead of column vectors [14]. In this scenario, a word-context matrix is created in which the context is given by words, phrases, sentences, or more clever possibilities, all derived from the analysis of a given corpus. In this type of matrix, context is represented by columns and target words are represented by rows.

The distributional hypothesis in linguistics mentioned before is the justification for applying the VSM to word similarity measurement. A word may be represented by a vector whose elements are derived from the occurrences of the word in various contexts, such as windows of words, grammatical dependencies or richer contexts proposals. Similar row vectors in the word-context matrix indicate similar word meanings.

When the vector elements values go beyond simple co-occurrence, by capturing syntactic relationships between words such as subject-verb, modifier-noun, etc., the word-context VSM becomes syntax-based. In this case, the context elements are generally formed by tuples *(r,w)* where w is a word occurring in relation type r with a target word t. The relations typically reflect argument structure (e.g., subject, object, indirect object) or modification (e.g., adjective-noun, noun-noun) and can be obtained via shallow syntactic processing [22] or full parsing [24]. The context elements *(r,w)* are treated as a single unit and are often called attributes or features.

To illustrate this type of VSM, a syntax-based word-context matrix consisting of four context elements and four target words is shown in Table 3.2.

Considering as corpus: *The truck might transport heavy rocks.*

Table 3.2: Syntax-based word-context matrix

Target words \ Features	(subj, truck)	(aux, might)	(mod,heavy)	(obj,rocks)
truck	0	0	0	0
transport	1	1	0	1
heavy	0	0	0	0
material	0	0	0	0

In this example, the matrix cells represent the number of times a target word t co-occurs with context elements *(r,w)*, as proposed in [24]. Because syntactic relationships capture more linguistic structure than word co-occurrences, they should at least in theory provide more informative representations of word meaning when used in word-context VSM.

Another approach to improve performance for measuring word similarity include word-context VSMs combined with the usage of a lexicon, such as WordNet [8]. Humans use both dictionary definitions and observations of word usage, so it is natural to expect the best performance from algorithms that use both distributional and lexical information [47]. This approach is applied in this work as part of our system structure (see section 4.2.1).

3.2 Semantic Space

Remembering the distributional characterization of semantics, whatever makes words similar or dissimilar in meaning is showed up distributionally in the lexical company of the word. A semantic space is a way of representing words as vectors in a Euclidean space with axes determined by a given linguistic context. A target word position with respect to other words expresses the degree of similarity between their meanings.

The semantic space could also be seen as a method of assigning each word in a language to a point in a real finite dimensional vector space. Formally it is a quadruple $<A,B,S,M>$ [26]:

- B is a set $b_{1...D}$ of context elements that determine the dimensionality D of the space and the interpretation of each dimension. B is often a set of words, but could be represented by a variety of linguistic forms.
- A specifies a lexical association function which will define the elements of a target word vector. So, each target word t is represented by a vector:

  $$v = [A(b_1, t), A(b_2, t), ..., A(b_D, t)]$$

  A may be the identity function.
- S is a similarity measure that maps pairs of vectors onto continuous valued quantity that represents similarity of meaning. Being the most popular similarity measures the Euclidean distance and the cosine.
- M is a transformation that takes one semantic space and maps it onto another, for example by reducing its dimensionality. M may also be an 'identity' mapping that does not change the space.

These four elements constitute formally a semantic space, however, it is fully functional just with B, A, and S specified. An important aspect during the semantic space construction is the context elements choice. When choosing them, there is a trade-off between choosing words that may not give reliable count statistics due to their low frequency and choosing high frequency words that provide reliable statistics but appear in almost every sentence of the language. In the first case, if only low frequency words are chosen as context elements then word vectors will be highly informative and distances in space will reflect nice distributional similarities; however, the semantic space will have high variance. In the second case, choosing very high frequency words seems reliable because high frequency words appear in nearly all sentences, however, word vectors will be similar because all words in the language tend to occur with the high frequency words and the semantic space will fail to reflect distributional differences. The ideal choice would include all words in the language, generating a very large vector. In practice this is not possible so a proper subset of words must be chosen, this is vital to get a good representation of words features.

Regarding the most popular similarity measures commonly used for S, the Euclidean distance and cosine formulas are expressed below:

- If the position of a point in a Euclidean n-space is seen as a vector, the Euclidean distance is the distance between two points. The formula is expressed in Equation 3.1.

$$d_E(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{3.1}$$

- The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them. The formula is expressed in Equation 3.2.

$$cos\theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \tag{3.2}$$

3.3 Similarities

A semantic space is suited to measure semantic relatedness. The semantic relatedness between two words a and b, $sem_r(a, b) \in \mathbb{R}$, depends on the degree of correspondence between the properties of a and b. The more correspondence exists, the greater their semantic relatedness.

Two words are semantically related if they have any kind of semantic relation [8]. This kind of relation is appreciated in synonyms, meronyms, antonyms, and words that are functionally related or frequently associated (e.g. chalk and blackboard). The term semantic relatedness in computational linguistics corresponds to attributional similarity in cognitive science [21].

A specific type of semantic relatedness in computational linguistics is the term semantic similarity applied just to words that share a hypernym. A hypernym, also called superordinate, is a linguistic term for a word whose meaning includes the meanings of other words (e.g. flower is hypernym of daisy and rose). So, while hypernyms are general words, hyponyms are subdivisions of more general words (e.g. daisy and rose are hyponyms of flower). These characteristics made semantic similarity term also to be known as taxonomical similarity.

Finally, there are two ways that words can be distributed in a corpus of text as defined in [41]. If two words tend to be neighbors of each other, then they are syntagmatic associates. If two words have similar neighbors, then they are paradigmatic parallels. Syntagmatic associates are often different parts of speech, whereas paradigmatic parallels are usually the same part of speech. Syntagmatic associates tend to be semantically associated (e.g. bee and honey often appear together); paradigmatic parallels tend to be taxonomically similar (e.g. engineer and technician have similar neighbors).

3.4 WordNet

WordNet is a large lexical database of English with existing versions for other languages [32]. It groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets. Synsets are interlinked by means of conceptual-semantic and lexical relations, lexical relations hold between word forms and semantic relations hold between word meanings. This results in a big network of meaningfully related words very useful for computational linguistics and natural language processing applications.

The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation). It links more general synsets (e.g. furniture) to increasingly specific ones (e.g. bed, chair). All noun hierarchies ultimately go up the root node 'entity'.

WordNet is commonly considered a thesaurus because it groups words together based on their meanings; however, there are important distinctions. First, WordNet labels the semantic relations among words and interlinks more than word forms, it includes a specific sense for every word. As a result, word sense disambiguation is avoided.

WordNet counts with a hierarchical semantic organization of its words, also called by computer scientists as "lexical inheritance system" where specific items inherit information from their generic superordinates. In other words, all of the properties of the superordinate are assumed to be properties of the subordinate as well. Figure 3.1 shows a part of WordNet graph to illustrate its hierarchical semantic organization.

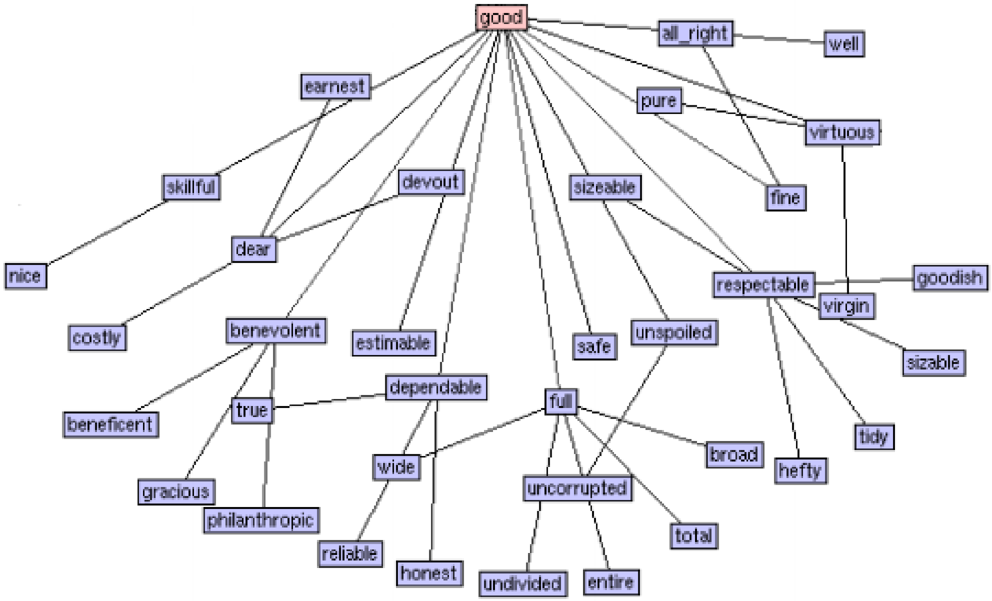


Figure 3.1: Part of WordNet graph from the point of view of the adjective 'good'

There are two forms to construe the hierarchical principle. The first one considers all nouns are contained in a single hierarchy. The second one proposes the partition of the nouns with a set of semantic primes representing the most generic concepts and unique beginners of different hierarchies [31]. As a result, WordNet adopted a set of twenty-five unique beginners that on the whole cover distinct conceptual and lexical domains. These set of semantic primes will be called in this work as WordNet top concepts and are shown in Table 3.3.

Table 3.3: WordNet unique beginners

List of 25 unique beginners for WordNet nouns	
{*act, action, activity*}	{*natural object*}
{*animal, fauna*}	{*natural phenomenon*}
{*artifact*}	{*person, human being*}
{*attribute, property*}	{*plant, flora*}
{*body, corpus*}	{*possession*}
{*cognition, knowledge*}	{*process*}
{*communication*}	{*quantity, amount*}
{*event, happening*}	{*relation*}
{*feeling, emotion*}	{*shape*}
{*food*}	{*state, condition*}
{*group, collection*}	{*substance*}
{*location, place*}	{*time*}
{*motive*}	

The structure of word strings when using WordNet in its electronic version is of the following form:

word#pos#sense

where *pos* indicate the part of speech of the word and *sense* is represented by an integer number to specify a determined word meaning, this allows avoiding word sense disambiguation problems.

WordNet also includes the implementation of semantic similarity and relatedness measures through a Perl module called *WordNet::Similarity* which implements a variety of similarity and relatedness measures based on statistical information found in the lexical database. Remembering the concepts defined in section 3.3, there is a remarkable difference between semantic similarity and semantic relatedness. A semantic relatedness measure uses all WordNet's relations for its calculation meanwhile a semantic similarity measure only uses the hyponymy relation.

In particular, *WordNet::Similarity* supports the measures of Resnik, Lin, Jiang-Conrath, Leacock-Chodorow, Lesk, Hirst-St.Onge, Wu-Palmer, Banerjee-Pedersen, and Patwardhan-Pedersen. However only three measures were considered in this work experimentation: Jiang and Conrath (JCN), Lin and the Lesk algorithm (Lsk). The first two are similarity measures which demonstrated to have a good performance among other measures that use WordNet as their knowledge source [9]; the last one is an adaptation of the original Lesk relatedness measure that takes advantage of WordNet's resources [2].

Jiang and Conrath - this measure combines the edge-based notion with the information content approach. The information content is commonly defined as $I(w) = -logP(w)$ where w is the word being measured. It calculates the conditional probability of encountering an instance of a child-synset given an instance of a parent synset, specifically their lowest super-ordinate (lso). This way the information content of the two words being measured, as well as that of their most specific subsume, influences the calculation. The formula is expressed in Equation 3.3.

$$dist_{JCN}(c_1, c_2) = 2\log(p(lso(c_1, c_2))) - (\log(p(c_1)) + \log(p(c_2))) \tag{3.3}$$

Lin - based on his similarity theorem: "The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are." It uses the same elements of JCN measure but in a different way. The formula is expressed in Equation 3.4.

$$sim_{LIN}(c_1, c_2) = \frac{2\log p(lso(c_1, c_2))}{\log p(c_1) + \log p(c_2)} \tag{3.4}$$

Lesk - the original algorithm measures the relatedness between two words by the overlap between their corresponding definitions as provided by a dictionary. Basically the steps are:

1. Retrieve from dictionary all sense definitions of the words to be measured.

2. Determine the definition overlap for all possible sense combinations.
3. Choose senses that lead to highest overlap.

But in WordNet an extended gloss overlap measure that combines the advantages of gloss overlaps with the structure of a concept hierarchy to create an extended view of relatedness between synsets is implemented [2].

3.5 Distributional Thesaurus

The word 'thesaurus' comes from the Greek language and means a storehouse or treasury of knowledge [36]. Soergel formally defines a thesaurus as "a list of terms and/or other signs (or symbols) indicating relationships among these elements" [45], in our case we view a thesaurus as a dictionary of related ideas reflected in a collection of concepts and terms.

A distributional thesaurus is a thesaurus generated automatically from a corpus by finding words which occur in similar contexts to each other [10]. This is not an easy task and different approaches for this type of information extraction had been studied [22]. One of them includes the usage of a similarity measure for the distributional thesaurus construction using a parsed corpus [24]. This work makes use of a system with these characteristics, so, a complete description of it is given in this section.

The process of measuring similarities between words according to their distribution in a text corpus consisted of different parts. First, dependency triples were extracted from the text corpus using a broad-coverage parser. A dependency triple consists of two words and the grammatical relationship between them in the input sentence, more specifically they are known as the head, the dependency type and the modifier. For example, the dependency triples in the sentence "I buy a black shirt" consist of:

(buy *subj* I), (buy *obj* shirt), (shirt *adj-mod* black), (shirt *det* a)

where *subj* is the relationship between a verb and its subject, *obj* is the relationship between a verb and its object, *adj-mod* is the relationship between a noun and its adjective modifier and *det* is the relationship between a noun and its determiner.

Dependency triples extracted from the corpus could be seen as features of the heads and modifiers in the triples. This idea is reflected in Table 3.4

showing a subset of the features of two nouns, as shown in [25]. Each row corresponds to a feature and the 'x' in the noun's column indicates belonging.

Table 3.4: Subset of features shared between two words: duty and sanction

Feature	"duty"	"sanction"	I(f)
subj-of(include)	X	X	3.15
obj-of(assume)	X		5.43
obj-of(avert)	X	X	5.88
obj-of(ease)		X	4.99
obj-of(impose)	X	X	4.97
adj-mod(fiduciary)	X		7.76
adj-mod(punitive)	X	X	7.1
adj-mod(economic)		X	3.7

Let *F(w)* be the set of features possessed by w. Then, *F(w)* can be seen as a description of the word w. The commonalities between two words w_1 and w_2 is then $F(w_1) \bigcap F(w_2)$. Having these elements, the similarity between two words is defined in Equation 3.5:

$$sim_{LIN}(w_1, w_2) = \frac{2 * I(F(w_1) \bigcap (F(w_2))}{I(F(w_1)) + I(F(w_2))} \tag{3.5}$$

where *I(S)* is the amount of information contained in a set of features S, its formula is expressed in Equation 3.6. Assuming that features are independent of one another,

$$I(S) = -\sum_{f \in S} log P(f) \tag{3.6}$$

where *P(f)* is the probability of a feature *f*. The probability *P(f)* can be estimated by the percentage of words that have a feature f among the set of words that have the same part of speech.

When two words have identical sets of features, they have complete similarity getting a maximum value of one. On the other hand, when two words do not have any common feature, they share no similarity getting a value of zero.

So, having a parsed corpus, the dependency triples are extracted and the pairwise similarity is computed between nouns, verbs and adjectives/adverbs that occurred at least n times (n common values are 50 or 100) using the similarity measure explained above. Then, for each word, a thesaurus entry containing the top-N (N is commonly $>$100) words that are most similar to it, is created. At the end, for every word w given as input of the distributional thesaurus, an output of the following format is delivered:

$$w : w_1, s_1; w_2, s_2; ...; w_N, s_N$$

where w_i is a word, $s_i = sim(w, w_i)$ and s_i values are ordered in descending order so that the most related words appear at the beginning of the list.

3.6 Latent Dirichlet Allocation

Let us recall the "bag of words" assumption where the order of words in a document does not need to be considered. Moreover, under this assumption, a specific order of documents in a corpus can also be neglected. This argument when transferred to probability theory is known as *exchangeability*.

The concept of exchangeability is expressed in de Finetti's theorem [13], also called de Finetti's representation theorem, and establishes that any collection of exchangeable random variables has a representation as a mixture distribution. So, if documents and words are considered as exchangeable representations, then mixture models that capture the exchangeability of both words and documents are necessary. This is the basic principle under the LDA model.

LDA is a generative probabilistic model for collections of discrete data such as a corpus [6]. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Traditional text categorization approaches consider a document belonging with a unique topic, while LDA assumes that a document may contain multiple topics where a topic is a multinomial distribution on words and a document is a multinomial distribution on topics as shown in Figure 3.2.

3.6.1 Statistical background on LDA

The following topics are intended to explain the statistic foundations under LDA in order to achieve a proper understanding about the model behavior.

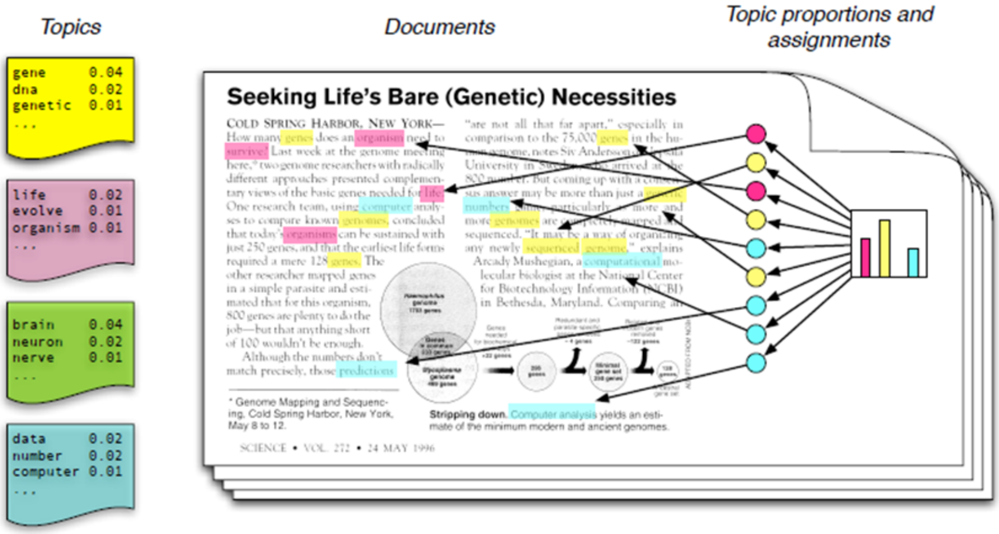


Figure 3.2: Generative model description of LDA. Picture taken from [6]

3.6.1.1 Bayes' Theorem

Bayes' theorem, also known as Bayes' law or Bayes' rule, shows the relation between two conditional probabilities that are the reverse of each other. It expresses the conditional probability, or posterior probability, of an event A after B is observed in terms of the prior probability of A, prior probability of B, and the conditional probability of B given A, denoted B/A.

The expression for the conditional probability of A given B provided by the Bayes' theorem is shown in Equation 3.7.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.7}$$

When applied to Bayesian models, B is replaced with observation y, A with parameter set Θ, and probabilities P with densities p. The denominator is dropped, which changes the relation from equal to 'proportional to', $\propto$. The model-based form is expressed in Equation 3.8.

$$p(\Theta|y) \propto p(y|\Theta)p(\Theta) \tag{3.8}$$

This form can be stated as the unnormalized joint posterior distribution,

$p(\Theta|y)$, being proportional to the likelihood, $p(y|\Theta)$, times the prior, $p(\Theta)$.

The posterior, $p(\Theta|y)$, is the result of updating prior information, $p(\Theta)$, with data, $p(y|\Theta)$. There are usually multiple parameters in a model, and together these create a joint distribution, which is why this is called the joint posterior. Lastly, it is unnormalized because the denominator of Bayes' theorem was discarded.

The likelihood, $p(y|\Theta)$, is the distribution of an unobserved variable y given data, and $p(\Theta)$ is the set of prior distributions of parameter set Θ before y is observed. So, the posterior, $p(\Theta|y)$, expresses uncertainty about Θ after taking the prior and data into account.

3.6.1.2 Prior Probabilities

A prior probability for a parameter is a description of what is known a *priori* about the parameter to be estimated. Bayesian inference considers prior probabilities and the data to estimate a resulting distribution, the posterior probability distribution.

Beyond this, there is a predecessor to prior probabilities called hyperprior used in hierarchical models. It is termed hierarchical because of the model's structure such that hyperpriors are used to estimate prior probabilities, which in turn are combined with the data to estimate the posterior probabilities.

Finally, a prior probability distribution, often called simply the prior, of an uncertain parameter θ or latent variable is a probability distribution that expresses uncertainty about θ before the data are taken into account. Prior distribution affects the posterior distribution.

3.6.1.3 Hierarchical Bayes

A hierarchical prior is a prior in which the parameters of the prior distribution are estimated from data via hyperpriors. Parameters of hyperprior distributions are called hyperparameters. Using hyperprior distributions to estimate prior distributions is known as hierarchical Bayes. In theory, this process could continue further. Estimating priors through hyperpriors, and from the data, is a method to obtain the optimal prior distribution. One of the uses for hierarchical Bayes is multilevel modeling.

Remembering that the unnormalized joint posterior distribution is proportional to the likelihood times the prior distribution.

$$p(\Theta|y) \propto p(y|\Theta)p(\Theta)$$

The simplest hierarchical Bayes model is expressed in Equation 3.9.

$$p(\Theta, \phi|y) \propto p(y|\Theta)p(\Theta|\phi)p(\phi) \tag{3.9}$$

Where ϕ is a set of hyperprior distributions. By reading the equation from right to left, it begins with hyperpriors ϕ, which are used conditionally to estimate priors $p(\Theta|\phi)$, which in turn are used to estimate the likelihood $p(y|\Theta)$, and finally the posterior distribution is $p(\Theta, \phi|y)$.

3.6.2 LDA Generative Model

LDA assumes the following generative process for each document w in a corpus D:

- Choose document size *N*.
- Choose distribution of topics $\theta \sim Dir(\alpha)$.
- For each of the N words w_n:
  - Choose a topic $z_n \sim Multinomial(\theta)$.
  - Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Where the topic variable z and the distribution of topics θ have a dimensionality *k*, the parameter $\sim$ represents a Dirichlet distribution which controls the mean shape and sparsity of θ, and β is a $k\ X\ V$ matrix parametrizing the word probabilities.

To enhance understanding about the generative process of LDA, the graphical model of LDA is shown in Figure 3.3 where nodes represent random variables, edges denote possible dependence, shaded nodes are observed variables and plates denote replicated structures. In this case, the outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

As shown in Figure 3.3, LDA is a three-level hierarchical Bayesian model. The parameters α and β_k are corpus-level parameters, assumed to be sampled once in the process of generating a corpus. The variables θ_d are document-level variables, sampled once per document. Finally, the variables Z_{dn} and W_{dn} are word-level variables and are sampled once for each word in each document.

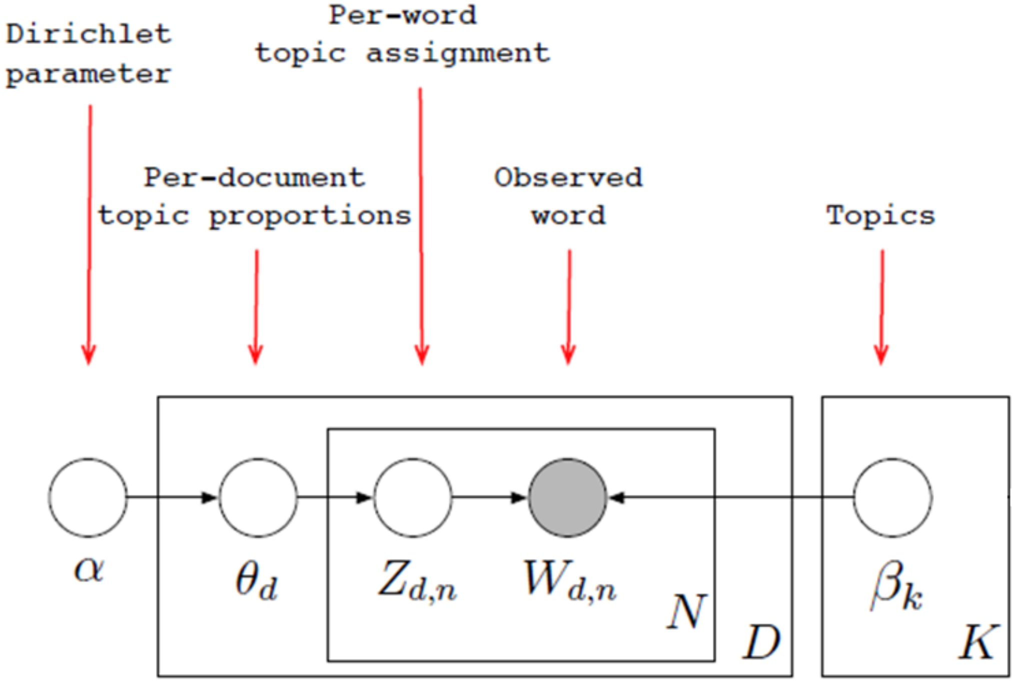


Figure 3.3: LDA graphical model

Also, the structure of the graph shown in Figure 3.3 defines the pattern of conditional dependence between the ensemble of random variables. The underlying joint distribution of the model is shown in Equation 3.10.

$$p(\theta, z, w|\alpha, \beta) = (\prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|z_{d,n}, \beta_{1:k})) \tag{3.10}$$

3.6.3 The Dirichlet Distribution

A Dirichlet distribution can be conceptualized as a probability distribution of probability mass functions (PMFs). Consider a pair of dice, each one has its own PMF (*i.e.* they have a different probability distribution of their sides) which can be slightly different between them. A Dirichlet distribution can be used to model the randomness of PMFs.

So, let $Q = [Q_1, Q_2, \ldots, Q_k]$ be a random PMF, that is $Q_i \geq 0$

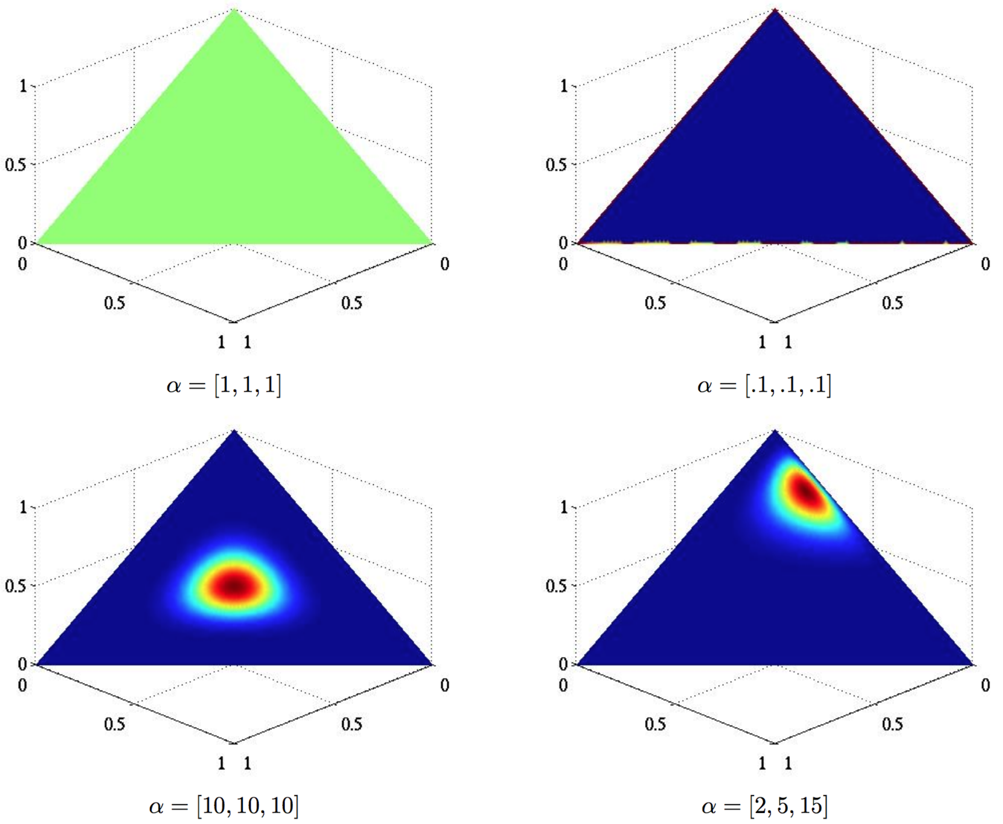
for $i = 1, 2, \dots, k$ and $\sum_{i=1}^{k} Q_i = 1$ (*i.e.* Q is a k-vector lying in the *(k-1)*-simplex). In addition, suppose a parameter $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k]$, with $\alpha_i > 0$ for each I, and let $\alpha_0 = \sum_{i=1}^{k} \alpha_i$. Then, Q is said to have a Dirichlet distribution with parameter α, denoted by $Q \sim Dir(\alpha)$, if its probability density function (PDF)>0 only on the simplex. The simplex is a *(k-1)*-dimensional object living in a k-dimensional space [19].

The Dirichlet distribution is parameterized by a vector α of positive real numbers. When all parameters $\alpha_1, \dots, \alpha_k$ of the Dirichlet distribution are equal, the PDF is symmetric around the middle. When each $\alpha_i < 1$, the probability is highest near the edge of the support. When each $\alpha_i > 1$, the probability is peaked at the middle of the support. When each $\alpha_i = 1$, the Dirichlet distribution is equal to a uniform distribution. Finally, when each α_i has a different value, the probability is skewed. Figure 3.4 shows an example of Dirichlet distribution given different values of α.

The Dirichlet distribution is often used as a prior in Bayesian statistics. When the Dirichlet distribution is used as a prior distribution for what is known about the parameters and the information got from the data followed a multinomial distribution, then, the posterior distribution again follows an (updated) Dirichlet distribution. In this case, the Dirichlet distribution became a conjugate prior for the multinomial distribution.

The Dirichlet has proved a good performance modeling the distribution of words in text documents [27]. Having a dictionary containing k possible words, then a particular document can be represented by a PMF of length k. A group of documents produces a collection of PMFs, and the Dirichlet distribution can be used to obtain the variability of the PMFs. Specific Dirichlet distribution applications are document modeling by different authors and document modeling by different topics such as LDA.

LDA operates in a space of distributions over words. Each distribution can be viewed as a point on the *(V-1)*-simplex, this could be known as the word simplex. The latent variable models consider k points on the word simplex and form a sub-simplex based on those points, in LDA the latent variables are the topics so this sub-simplex could be known as the topic simplex. Any point on the topic simplex is also a point on the word simplex, as seen in Figure 3.5. So, LDA proposes that each word of both the observed and unseen documents is generated by a randomly chosen topic which is drawn from a distribution with a randomly chosen parameter. This parameter (θ) is sampled once per document from a smooth distribution (α) on the topic simplex.



$\alpha = [1, 1, 1]$

$\alpha = [.1, .1, .1]$

$\alpha = [10, 10, 10]$

$\alpha = [2, 5, 15]$

Figure 3.4: Density plots of Dirichlet distributions over the probability simplex in a three dimensional space using different values of α, with low densities represented by blue color and high densities represented by red color. Picture from [19]

3.6.4 LDA Inference

The inferencial problem in LDA consists on computing the posterior distribution of the hidden variables given a document, this is expressed in Equation 3.11.

$$(p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{3.11}$$

A variety of approximate inference algorithms are considered for LDA,

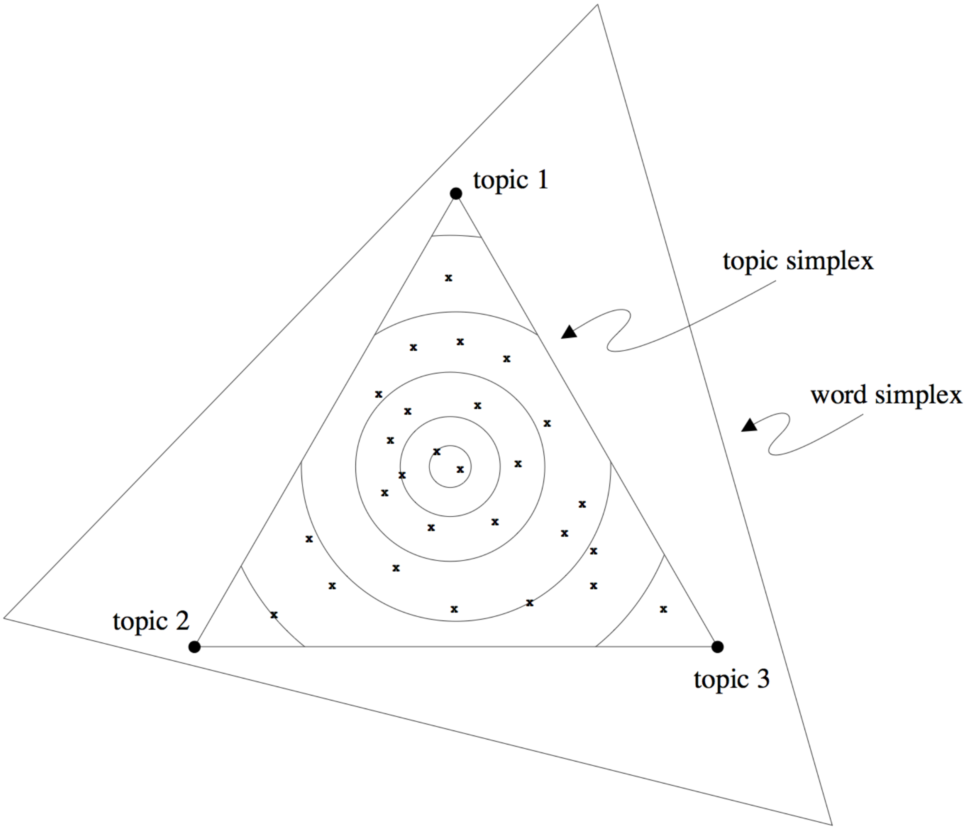


Figure 3.5: The topic simplex for three topics embedded in the word simplex for three words. Each corner of the word simplex corresponds to the distribution where each respective words has probability one. The three points of the topic simplex correspond to three different distributions over words. LDA places a smooth distribution on the topic simplex denoted by the contour lines. Picture from [6].

being Gibbs Sampling one of the major approaches to inference in complex probabilistic models. Gibbs sampling approaches the underlying distribution by sampling a subset of variables conditional on fixed values of all other variables [7].

Let $X = (X_1, \dots, X_n)$ be a random vector considered a Markov chain with states $x = (x_1, \ldots, x_i, \ldots, x_n) \in A$ for some set A. The Gibbs sampler for generating a random vector X conditional on the event that $X \in A$ moves from state to state by choosing a coordinate I at random and then generating a random variable from the conditional distribution of X_I given the values of the other random variables, $X_j, j \neq I$. If the vector obtained from replacing the old value of X_I by this generated value remains in A, then it becomes the next state, and if not, then the next state remains unchanged from the previous one [38].

The basic idea in Gibbs sampling is that, rather than probabilistically picking the next state all at once, a separate probabilistic choice for each of the k dimensions is done, where each choice depends on the other *k-1* dimensions. Each dimension corresponds to a parameter or a variable of the model [37].

So, Gibbs sampling makes it possible to compute expected values, by defining a conceptually straightforward approximation. This approximation is based on the idea of a probabilistic walk through a state space whose dimensions correspond to the variables or parameters in the model.

After explaining the LDA algorithm, the topics required to understand the search-by-concept dictionary operation are completed. The following chapter describes the methodology implemented to carry out the reverse search of the dictionary system proposed in this research.

Chapter 4

Methodology

Once general concepts needed to understand the system operation were explained, a complete description of the methodology implemented in this project is given in this chapter. First, the procedure of word vectors creation is detailed for each one of the different sources selected to build a semantic space. Then, the reverse search process carried out in the semantic space is explained together with an example. This way, all the characteristics involved on the search-by-concept dictionary operation are shown.

4.1 Introduction

A reverse dictionary receives a definition as input and gets a word representing that definition as output. The search-by-concept dictionary proposed in this project is based on this principle. One of the contributions of this system is the interpretation given to the input data, which for a reverse dictionary are the words forming the input concept. In this work, every word is represented as a vector in order to constitute a Semantic Space and three different sources are proposed to determine the numeric values of the vector elements. As a consequence, three different Semantic Spaces are constructed, each one with determined properties mentioned through this chapter.

Semantic Spaces are the key for a successful reverse lookup in the proposed system. Exploiting the vector-space word representation advantages, such as the level of generalization achieved on every word due to their distributed representation, its capacity to capture semantic regularities in language and the proximity in space given to similar words; an algebraic analysis between vectors is done in order to select a sample of candidate words conditioned on some parameters. Finally, a predefined number of output words is displayed to the user.

This is an overall picture of the system performance aiming to explain the structure in the chapter's content, giving more clarity during its reading.

4.2 The Semantic Space

A semantic space is a way of representing words as vectors in a Euclidean space with axes determined by a given linguistic context. Three different sources were proposed for semantic space construction, having different linguistic contexts for word vectors. In this section is described how word vectors were constructed depending on the following sources:

- WordNet - a large lexical database of English (see Section 3.4).
- Distribution thesaurus - a thesaurus generated automatically from a corpus by finding words occurring in similar contexts to each other (see Section 3.5).
- Latent Dirichlet Allocation - a topic modeling algorithm (see Section 3.6).

With these three sources, the dimensionality of each semantic space could be fixed.

4.2.1 WordNet as a Source for Semantic Space Construction

Under the semantic space based on WordNet lies a semantic analysis of words using semantic similarity and relatedness measures to represent their vectors. When using this kind of measures on an ontology-structured resource such as WordNet, it is only possible to calculate semantic similarity or semantic relatedness between words that belong to the same part of speech. Due to this part of speech restriction, only nouns are considered as word members of the semantic space. Besides, it is well known that in natural language, concepts are expressed mostly as noun phrases [46].

WordNet on its version 3.0 includes 82115 synsets where 117798 nouns are distributed. The structure of word strings in WordNet describes a specific sense of a certain word as shown below; this is used to avoid word sense disambiguation problems:

word#pos#sense

where *pos* refers to the part of speech of the word and its sense is represented by an integer number.

Based on WordNet's hierarchical principle (see Section 3.4), the 25 top concepts were defined as semantic primes to represent the linguistic context that determine the dimensionality of the space. The top concepts with their specific senses are listed below:

activity#n#1 *animal#n#1* *artifact#n#1* *attribute#n#2*
body#n#1 *cognition#n#1* *communication#n#2* *event#n#1*
feeling#n#1 *food#n#1* *group#n#1* *location#n#1*
motive#n#1 *natural_object#n#1* *natural_phenomenon#n#1* *human_being#n#1*
plant#n#2 *possession#n#2* *process#n#6* *quantity#n#1*
relation#n#1 *shape#n#2* *state#n#1* *substance#n#1*
time#n#5

This is also the order given to the top concepts during vector representation of words mentioned further on.

So, for every WordNet noun, its vector was created calculating the semantic similarity or relatedness value between the respective noun and each top concept. Three measures were considered: JCN, Lin and the Lesk algorithm. The first two are similarity measures and the last one is a relatedness measure. After reading and creating the vectors for every noun, the process ends. The algorithm describing this process is expressed in Algorithm 1.

A more detailed description of the process evolution is shown in Figure 4.1. The process was repeated for each of the different measures mentioned above, resulting on a semantic space with JCN measured vectors, another with Lin measured vectors, and the last one with Lsk measured vectors.

With the semantic spaces created, a normalization procedure was performed to adjust the maximum value of vector elements to 1 and to fix a precision of five decimal places. The maximum values of each dimension used to normalize word vectors are shown in Table 4.1 for the three measures of semantic distance proposed.

Table 4.1: Maximum values used to normalize WordNet semantic space

Top concept \ Measure	JCN	LIN	LSK
activity#n#1	1.285089	1	1.342394
animal#n#1	2.190239	1	1.960529
artifact#n#1	1.050768	1	1.345518
attribute#n#2	1.297712	1	1.462529
body#n#1	1.177915	1	1.620097
cognition#n#1	1.619827	1	1.276847
communication#n#2	1.034699	1	1.338452
event#n#1	2.854946	1	1.41939
feeling#n#1	1.07354	1	1.462761
food#n#1	0.896673	1	2.281172
group#n#1	1.527866	1	1.259256
location#n#1	1.622349	1	1.621083
motive#n#1	1.691525	1	1.638278
natural_object#n#1	0.669271	1	1.213891
natural_phenomenon#n#1	7.26094	1	1.26911
human_being#n#1	6.709807	1	3.267898
plant#n#2	9.051771	1	1.960529
possession#n#2	1.320277	1	1.342498
process#n#6	2.44243	1	1.210678
quantity#n#1	1.361789	1	1.209628
relation#n#1	1.789515	1	1.462529
shape#n#2	0.812525	1	1.268193
state#n#1	2.172894	1	2.83656
substance#n#1	3.582165	1	1.558951
time#n#5	0.875268	1	1.378288

Algorithm 1 WordNet semantic space construction.

```
Input: WordNet DB
for all WordNet words do
    Read a new word $w$
    if $w$ is a noun then
        for all $tc_i \in top\_concepts$ do
            Calculate semantic similarity between $w$ and $tc_i$
            Define the value calculated as the element $[i]$ of $w$ vector $v$
        end for
        Save $v$ in WordNet semantic space.
    end if
end for
```

In the case of Lin measure, the normalization procedure was not necessary because the maximum value using Lin semantic similarity measure is 1 (see Equation 3.5 explanation for recalling terms); so, vectors had already the desired structure.

Finally, word vectors having the following form were obtained:

```
genius#n#1 ->
0.05748, 0.04058, 0.09603, 0.06138, 0.06117, 0.04774, 0.07306, 0.02822,
0.06301, 0.07750, 0.05024, 0.05693, 0.03530, 0.12316, 0.01008, 0.01046,
0.00898, 0.05117, 0.03144, 0.05603, 0.04203, 0.07932, 0.03364, 0.02163,
0.07081
```

Having vectors in this form, the WordNet semantic space construction was completed and word vectors were ready to conduct a reverse search.

4.2.2 Distributional Thesaurus as a Source for Semantic Space Construction

The distributional thesaurus used in this project contains the pairwise similarity between 4808 nouns and for each one, a thesaurus entry containing the top-200 words that are most similar to it was created. The similarity value between words was computed using Lin similarity measure (see Section 3.5).

The semantic space construction based on this resource goes further in comparison with WordNet's. Two manners for vector representation of words were proposed, one that preserves the topics used for WordNet semantic

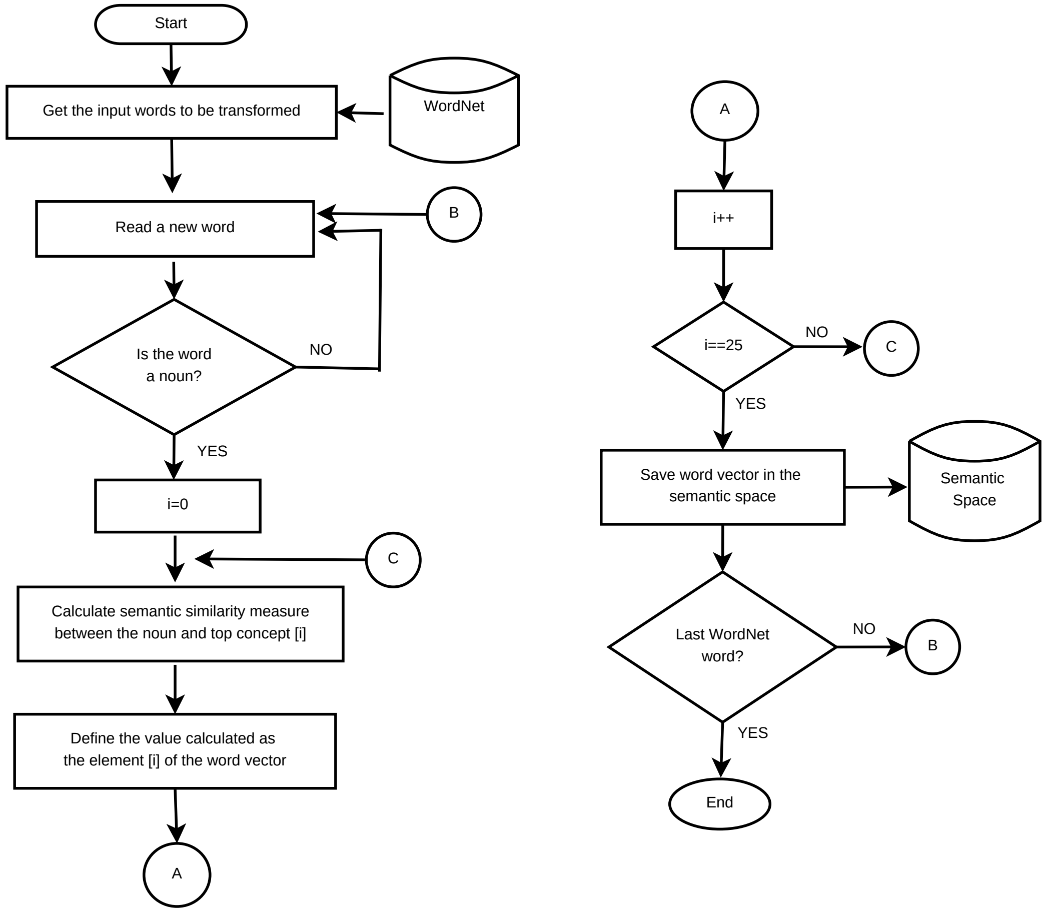

Start
Get the input words to be transformed
WordNet
Read a new word
B
Is the word
a noun?
NO
YES
i=0
C
Calculate semantic similarity measure
between the noun and top concept [i]
Define the value calculated as
the element [i] of the word vector
A
A
i++
i==25
NO
C
YES
Save word vector in the
semantic space
Semantic
Space
Last WordNet
word?
NO
B
YES
End

Figure 4.1: WordNet semantic space construction flowchart

space and a new dynamic distribution of topics. Regardless the vector representation type, the initial step in the semantic space construction was the vocabulary extraction from the pairwise similarity values database.

Regarding the first type of vector representation, each word of the vocabulary was represented as a vector of 25 dimensions determined by the 25 top concepts in WordNet's graph [31]. These topics were selected due to their character of semantic primes representing the most generic concepts and unique beginners of different hierarchies, earning an equilibrated distribution of vector's dimensions. However, the specific sense of each topic could not be considered due to the corpus structure. The top concepts are listed below:

activity *animal* *artifact* *attribute* *body*
cognition *communication* *event* *feeling* *food*
group *location* *motive* *natural object* *natural phenomenon*
human being *plant* *possession* *process* *quantity*
relation *shape* *state* *substance* *time*

Consequently, a vector dimension value consists of the semantic similarity measured between the word being analyzed and each top concept. If there was no similarity value defined in the pairwise similarity database, the dimension value was zero. The similarity values depend on the number of features shared between the words being analyzed, so if two words do not have any common feature, then their similarity value is zero. A word vector containing zeros in all its dimensions was discarded from the semantic space. Word vectors of this form constituted the Thesaurus semantic space I (TSSI). The algorithm describing this kind of vector representation is shown in Algorithm 2 and its flowchart is detailed in Figure 4.2.

Algorithm 2 TSSI construction.

```
Input: Distributional Thesaurus DB
Vocabulary extraction
for all nouns do
    Read a new noun $n$
    for all $tc_i \in top_concepts$ do
        if pairwise similarity value $psv$ between $n$ and $tc_i$ exists then
            Define $psv$ as the element $[i]$ of $n$ vector $v$
        else
            Define zero as the element $[i]$ of $n$ vector $v$
        end if
    end for
    if $psv$ is not a vector full of zeros then
        Save $v$ the Thesaurus semantic space I
    end if
end for
```

During the TSSI construction, a vocabulary loss was noticed due to the number of words with vectors full of zeros. The discarded words could have

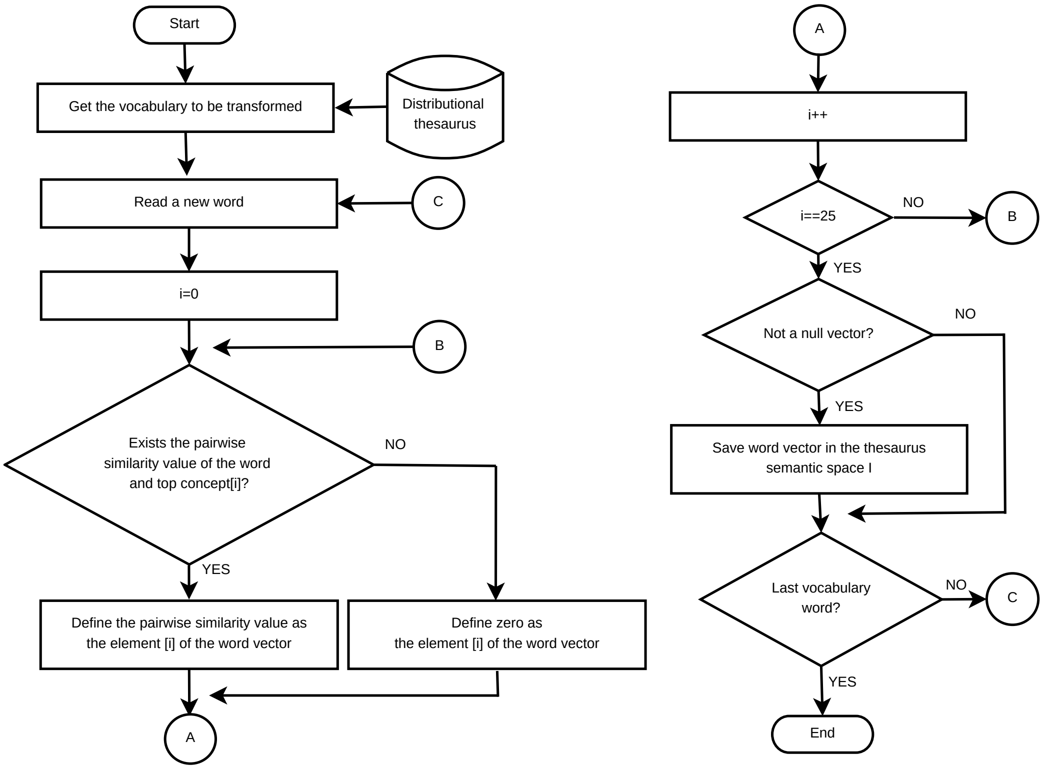

Start
Get the vocabulary to be transformed
Distributional thesaurus
Read a new word
C
i=0
B
Exists the pairwise similarity value of the word and top concept[i]?
NO
YES
Define the pairwise similarity value as the element [i] of the word vector
Define zero as the element [i] of the word vector
A
A
i++
i==25
NO
B
YES
Not a null vector?
NO
YES
Save word vector in the thesaurus semantic space I
Last vocabulary word?
NO
C
YES
End

Figure 4.2: TSSI construction flowchart

been good answers during testing, but this assumption depends totally on the input concept. In order to keep the maximum number of words related to our dictionary input concept, a dynamic topic generation was proposed. This is the second manner for the vector representation of words.

For every input concept in the dictionary, a set of topics was generated with the most related terms of each noun included in the input concept. The set of topics represents the linguistic context determining the dimensionality of the space and received the name of dynamic topics. The dimensionality of the vectors depended on the number of nouns forming the input concept and could not be over 25. So, for each input concept of n nouns, the highest number k satisfying the expression $k \cdot n \leq 25$ was calculated. Then, k became the number of the most related terms selected for each noun member of the input concept, getting at the end a set of topics related at least with one of the members of our input and discarding from the semantic space only those words with no relation at all, getting a loss of vocabulary that does not affect the quality of the reverse dictionary results. The algorithm corresponding to the dynamic topic generation is shown in Algorithm 3 and its flowchart is detailed in Figure 4.3.

Algorithm 3 Dynamic topics generation.

```
Input: Concept for reverse search
Get the number of nouns $N$ forming the input concept
Define the highest possible value $k$ satisfying $k * N \leq 25$
for all $n_i \in nouns$ do
    Get the $k$ most related terms $mrt$ of $n_i$ from the distributional thesaurus database
    Store $mrt$ in the dynamic topic set $dts$
end for
```

Once the dynamic topics were obtained, the TSSI construction algorithm was applied. The only difference is that top concepts were replaced by dynamic topics. Finally, word vectors were saved in the thesaurus semantic space II (TSSII).

For example, supposing the following input concept: *"ball field stick sport"*

$$n = 4; k = 6$$

For each noun member of the input, their six most similar terms are extracted from the Thesaurus DB being the following:

```
ball - puck, shot, pass, pitch, fastball, bat
field - area, sector, industry, floor, forest, land
stick - knife, bat, baton, machete, broomstick, glove
sport - baseball, soccer, football, basketball, boxing, golf
```

The joint of all similar terms would represent the linguistic context determining the dimensionality of the semantic space subsequently constructed. In the case that the extracted terms are shared between input nouns, only one is taken into account, as a consequence, the dimensionality of the semantic space is reduced.

Again, a normalization procedure was not necessary due to Lin's semantic similarity measure properties. With the semantic spaces construction completed, word vectors were ready to conduct a reverse search.

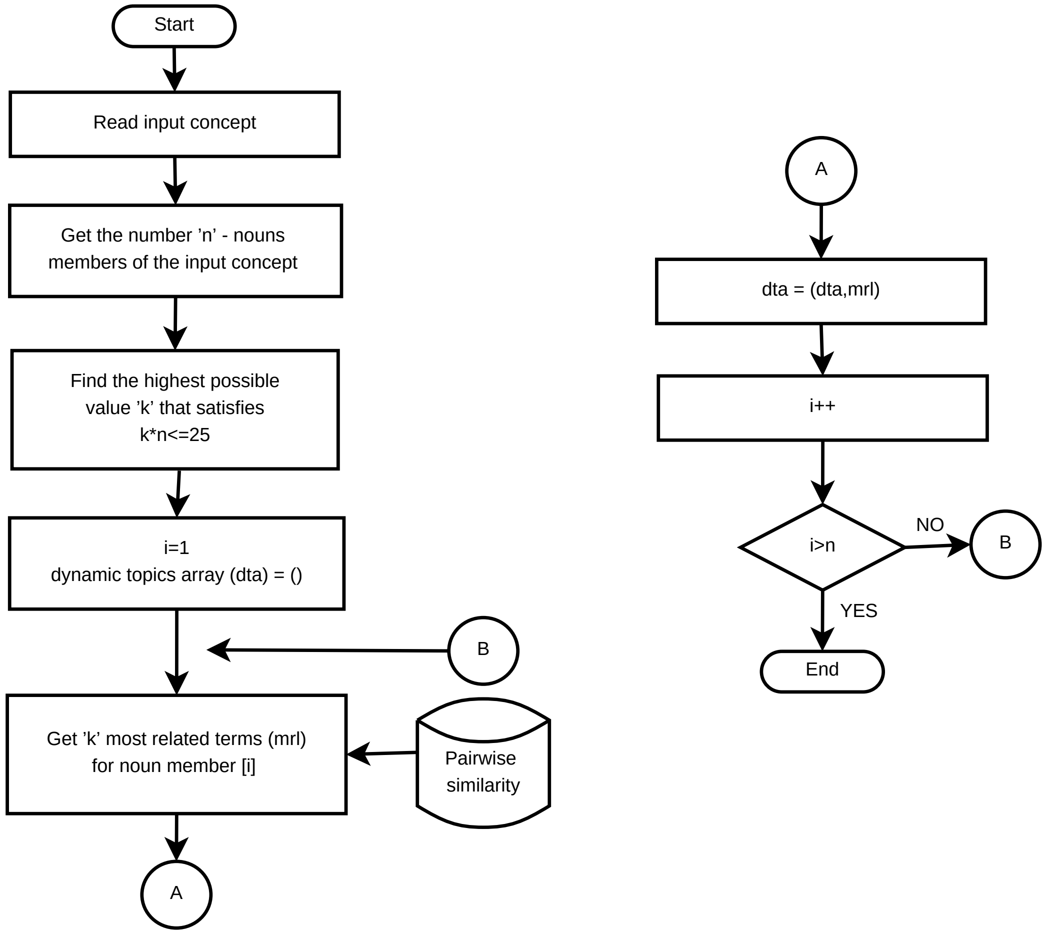

Start
Read input concept
Get the number 'n' - nouns members of the input concept
Find the highest possible value 'k' that satisfies k*n<=25
i=1
dynamic topics array (dta) = ()
B
Get 'k' most related terms (mrl) for noun member [i]
Pairwise similarity
A
A
dta = (dta,mrl)
i++
i>n
NO
B
YES
End

Figure 4.3: Dymanic topic generation flow chart

4.2.3 Latent Dirichlet Allocation as a Source for Semantic Space Construction

In order to generate word vectors using LDA, it was necessary a corpus to be modeled. The corpus selected for this task was the Wikipedia corpus which includes 4,105,489 articles in English and a vocabulary of 7,423,153 words. Wikipedia is a multilingual, web-based, free-content encyclopedia that covers a wide variety of topics making it an extraordinarily large corpus with broad scope.

During corpus processing it was noticed that a large number of terms appeared just a few times among all articles (for example, *"AAABBNNNNN"* appearing once). Such terms would not give relevant information during word vector representation; so, words appearing less than 5 times in the corpus were removed. After removing stopwords and words appearing less than 5 times, the vocabulary of the corpus was reduced to 1,680,882 words.

So, the Daichi Mochihashi LDA package [34] was implemented using the Wikipedia corpus in a specific data format. The data file needed to be a text file, with each line representing a Wikipedia article. A typical data file had the following form:

```
1:1 3:2 6:2
1:2 5:1 3:3 2:1 4:1
2:4 1:1 6:2
```

- Each line could be maximum 65535 bytes (about 820 lines in 80-column text).
- Each line consisted of pairs *<word_ id>:<count>*. Where *word_ id* consisted of an integer from 1 to V (with V being the vocabulary size), and *count* consisted of an integer representing the number of times that word appeared in the article.
- *<word_ id>:<count>* pairs needed to be separated by white spaces.

With the Wikipedia corpus ready, the remaining parameter was the number of latent topics T to be assumed in the data. An assumption of 100 topics was made following traditional selection choices [6].

After implementing LDA, two outputs were generated:

- α - A T-dimensional row vector representing the parameter of prior Dirichlet distribution over the latent topics.
- β - A *[V, T]*-dimensional matrix representing the set of words for each latent topic where V is the size of the vocabulary.

Analyzing the output obtained, each cell in β matrix indicates the probability of a specific word v_i with each one of the topics automatically generated by LDA, where $v_i \in V$. Thus, the rows from β matrix could be seen as word vectors of 100 elements as in a traditional word context matrix (see Section

3.1.2). Those word vectors constituted the LDA semantic space in which the linguistic context determining the axes of the space was represented by the untagged topics automatically generated since LDA is an unsupervised method.

4.3 The Reverse Search Process

Based on the reverse dictionary basic principle, this section explains the procedure used to get a list of target words given an input concept. The reverse search overview is shown in Figure 4.4.

The system input consists of a concept formed of n nouns. After reading the n nouns, the system looks for their respective vectors in the semantic space previously created by the user and begin with the semantic space analysis. This analysis varies depending on the semantic space selected but always begins calculating the average vector resulting from the input words, giving as a result a new vector that should be located in the semantic space representing a word combining the semantics of the nouns members of the input concept. However, getting an average vector located exactly over an existing word in the semantic space is highly unlikely; thus, a sample of the N-nearest neighbors is taken (N may vary depending on the Euclidean distance parameter).

So, according to the semantic space source, different parameters were considered for the selection of the N-nearest neighbors. A part of it would belong to the system output, in that sense, the parameters were proposed to improve the output quality by taking advantage of the semantic spaces properties. These are listed below:

For the semantic space based on WordNet, two parameters were considered:

1. The Euclidean distance between vectors needed to be:

    - For JCN less than 0.1
    - For Lin less than 0.8
    - For Lsk less than 0.1

    These threshold values were determined after extensive testing. It was noticed that vectors with Euclidean distances bigger than the values

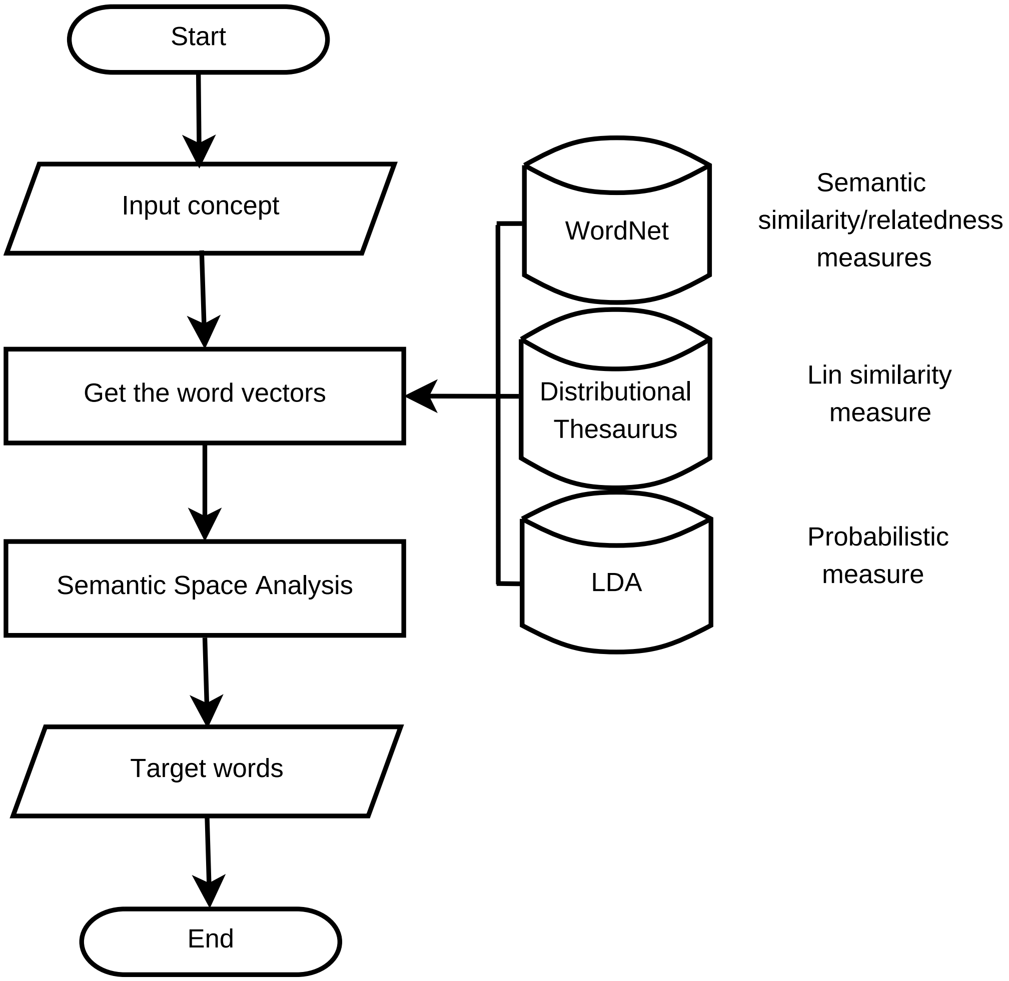


Figure 4.4: Reverse search overview

mentioned above tend to represent words having no relationship with the input concepts.

2. The product of the semantic similarity measured between each member of the input and the word represented by the candidate vector was computed; the top t words with the highest values were chosen to form the system output (being t specified by the user). This was considered the ranking phase.

For the semantic space based on the distributional thesaurus, these were the parameters considered:

1. The Euclidean distance between vectors needed to be:

   - For TSSI less than 0.3
   - For TSSII less than 0.5

   Again, these threshold values were determined after extensive testing. It was noticed that vectors with Euclidean distances bigger than the values mentioned above tend to represent words having no relationship with the input concepts.

2. Candidate words having a pairwise similarity value defined in the distributional thesaurus DB with each noun member of the input concept have priority over other candidates.

3. The product of the pairwise semantic similarity values between each noun member of the input concept and the candidate words is calculated; the top t words with the highest values were chosen to form the system output (being t specified by the user). This was considered the ranking phase.

Finally, for the semantic space based on LDA, only one parameter was considered:

1. The Euclidean distance between the average vector and the words existing in the semantic space. The top t words with the shortest distances were chosen as the target words displayed in descending order (being t specified by the user).

The reason why in LDA semantic space could not be used a second parameter for ranking is the nature of the algorithm. Being an unsupervised source, the creation of vectors was completely automatic and additional data with the possibility of being used in a ranking phase was not available.

With the complete description of the parameters used in every semantic space to determine the output words, the reverse search process ends as well as the contents of this chapter.

Chapter 5

Experimentation and Results

In this chapter the experiments performed following the methodology in Chapter 4 are described together with the results obtained for each one of the semantic spaces proposed. After the analysis of results, in order to evaluate the system's performance, a proposal to determine which model is the closest to human associative reasoning is explained and the results of an existing electronic reverse dictionary are taken into account for comparison terms. Finally, the results of the evaluation are presented in different tables and graphics.

5.1 Experimentation

In order to carry out experiments on the search-by-concept dictionary proposed in this work, a test set with 50 different concepts was created. Each concept is formed of n nouns, the value of n was determined based on the state of the art [15][51][43], so concepts having $2 < n \leq 4$ were considered. However, our system allowed input concepts formed of multiple nouns.

The complete test set is shown in Table 5.1. The concepts' structure varied between sources of vector representation; while the Distributional Thesaurus and LDA keep the structure mentioned in Table 5.1, for WordNet was necessary to specify the part of speech and sense of each noun member of the concept due to WordNet properties (see Section 3.4).

Table 5.1: Experimentation test set of 50 concepts

Dictionary input concepts
1. glory flag battlefield
2. genius poetry “love affair”
3. music dancing stage
4. chain battlefield troop
5. criminal corruption mafia
6. dark talk detective
7. motor wheel driver
8. nature evolution life
9. intelligence technology profession
10. classroom student professor
11. swimsuit hotel sand
12. alcohol cigarette drug
13. blood punch sport
14. cake balloon candy
15. stadium grass player
16. antenna screen broadcast
17. wheel motor “steering wheel”
18. “gm shoe” “athletic contest” race
19. cement rod sand
20. string tune tuning “musical instrument”
21. computer noise security
22. candle priest christian
23. furniture door window
24. recipe ingredient oven
25. flour oven fire
26. bomb weapon battle
27. hair mirror comb
28. bag clothing travel
29. bed syringe nurse
30. cadaver coffin crying sadness
31. wood nail “power saw”
32. costume dancing mask ballerina
33. star orbit planet
34. thunderbolt cloud water
35. satellite antenna transmission
36. church pope Rome
37. clown laugh show
38. car building people
39. bacteria disease cold
40. cattle barn cow
41. field ball stick sport
42. sight smell taste
43. brother grandparent cousin
44. filming script actor
45. hymn flag emblem
46. foam soap water clothing
47. bone pain crack
48. discipline map earth
49. birthday happiness surprise
50. computer data science

5.1.1 WordNet

Word vectors based on WordNet had three different types: JCN, Lin and Lsk. To determine which semantic similarity (JCN and Lin) or relatedness Lsk measure had the best quality output, the complete test set was implemented on each semantic space. To describe the search-by-concept process carried out for each dictionary input, an example for JCN semantic space is explained:

```
Input concept - gym_shoe#n#1 athletic_contest#n#1 race#n#2

gym_shoe#n#1 ->
0.05383, 0.03492, 0.11093, 0.05720, 0.05738, 0.04458, 0.06833,
0.02628, 0.05933, 0.07286, 0.04694, 0.05249, 0.03347, 0.11451,
0.00944, 0.00927, 0.00782, 0.04819, 0.02938, 0.05237, 0.03932,
0.07490, 0.03153, 0.02020, 0.06700

athletic_contest#n#1 ->
0.08950, 0.03136, 0.07729, 0.07229, 0.05214, 0.06832, 0.08528,
0.04630, 0.07227, 0.07297, 0.05879, 0.04805, 0.04601, 0.10280,
0.00946, 0.00849, 0.00708, 0.05869, 0.02943, 0.06550, 0.04902,
0.09036, 0.02934, 0.02281, 0.08023

race#n#2 ->
0.09333, 0.03214, 0.07960, 0.07480, 0.05331, 0.07113, 0.08805,
0.04859, 0.07433, 0.07472, 0.06073, 0.04942, 0.04732, 0.10539,
0.00970, 0.00866, 0.00724, 0.06035, 0.03020, 0.06766, 0.05061,
0.09279, 0.03003, 0.02350, 0.08229

Average vector ->
0.07888, 0.03280, 0.08927, 0.06809, 0.05427, 0.06134, 0.08055,
0.04039, 0.06864, 0.07351, 0.05548, 0.04998, 0.04226, 0.10756,
0.00953, 0.00880, 0.00738, 0.05574, 0.02967, 0.06184, 0.04631,
0.08601, 0.03030, 0.02217, 0.07650
```

After the semantic space analysis specifying a $t=7$ the highest ranked output words are shown in Table 5.2 where the most relevant result was *meet#n#1*. The proximity of its vector's dimensions values with the ones of the average vector previously calculated is remarkable.

```
meet#n#1 ->
0.08617, 0.03065, 0.07523, 0.07008, 0.05108, 0.06587, 0.08282,
0.04433, 0.07043, 0.07140, 0.05706, 0.04682, 0.04483, 0.10047,
0.00925, 0.00833, 0.00693, 0.05719, 0.02873, 0.06359, 0.04762,
0.08818, 0.02873, 0.02220, 0.07838
```

Table 5.2: Dictionary output for concept: gym_shoe#n#1, athletic_contest#n#1, race#n#2. WordNet Semantic Space

Product of semantic similarity values	**Euclidean distance**	**Word**	**Gloss**
0.02642	0.02015	meet#n#1	a meeting at which a number of athletic contests are held
0.0058	0.02755	Olympic_Games#n#1	the modern revival of the ancient games held once every four years in a selected country
0.00426	0.02755	horse_race#n#1	a contest of speed between horses
0.00426	0.02755	footrace#n#1	a race run on foot
0.00387	0.05936	game#n#2	a single play of a sport or other contest
0.00325	0.03846	track_meet#n#1	a track and field competition between two or more teams
0.00293	0.04428	race#n#1	any competition

Also, it is notable in Table 5.2 the weight given to the parameter represented by the product of semantic similarity values, being the key factor for the output word selection. Even words with lower Euclidean distances from the average vector are displaced due to the impact of the product of semantic similarity. This allows discarding words that could be located near to the average vector but having no relation with the input concept, becoming an irrelevant candidate. And with the Euclidean distance threshold for each semantic space (see Section 4.3) it is ensured in every case output words with close proximity to the average vector.

So, the experiment mentioned above was repeated for each semantic space; then, with every input concept included in the test set. Table 5.3 shows the reverse search of three different concepts with the two highest ranked output words from each semantic space. Notice that when working with WordNet it was possible to include the gloss of each word improving the output with additional information. Also, it could be seen in some cases that output words may be shared between source measures of vector creation, specially between JCN and Lin measure, this may have its origins in their respective formulas of semantic similarity since Lin measure uses the same elements of JCN measure but in a different way. This was a constant behavior through the test set results.

Table 5.3: Reverse search of three different concepts - WordNet Semantic Spaces

Concept	**System results**		
nature evolution life	JCN	growth#n#2	A progression from simpler to more complex forms.
		chemical_reaction#n#1	(Chemistry) a process in which one or more substances are changed into others.
	Lesk	oxidative_phosphorylation#n#1	An enzymatic process in cell metabolism that synthesizes ATP from ADP.
		blooming#n#1	The organic process of bearing flowers.
	Lin	growth#n#2	A progression from simpler to more complex forms.
		heat_sink#n#1	A metal conductor specially designed to conduct (and radiate) heat.
antenna screen broadcast	JCN	serial#n#1	A serialized set of programs.
		wide_screen#n#1	A projection screen that is much wider than it is high.
	Lesk	rerun#n#1	A program that is broadcast again.
		receiver#n#1	Set that receives radio or tv signals.
	Lin	electrical_device#n#1	A device that produces or is powered by electricity.
		surface#n#1	The outer boundary of an artifact or a material layer constituting or resembling such a boundary.
thunderbolt cloud water	JCN	atmospheric_electricity#n#1	Electrical discharges in the atmosphere.
		precipitation#n#3	The falling to earth of any form of water.
	Lesk	atmospheric_electricity#n#1	Electrical discharges in the atmosphere.
		cumulus#n#1	A globular cloud.
	Lin	atmospheric_electricity#n#1	Electrical discharges in the atmosphere.
		atmospheric_phenomenon#n#1	A physical phenomenon associated with the atmosphere.

At first sight, the results obtained from each semantic space seem to be correct answers for each concept, however, it was necessary to determine which semantic similarity or relatedness measure implemented for word vector creation had the best performance in reverse search. To accomplish it, complete results of each semantic space were evaluated as detailed in [33] under two considerations:

1. Indicate if the output words converged with their associative reasoning.
2. Indicate the source measure of vector creation that gave the best result.

It was concluded that the output words with JCN-measured vectors were the best combining the semantics of the input concepts. This implied that WordNet semantic space would be represented by JCN-measured vectors during the general evaluation.

5.1.2 Distributional Thesaurus

The distributional thesaurus was the second source for semantic space construction having two proposals, TSSI and TSSII, each one containing word vectors with different characteristics. The first one based on WordNet top concepts was precreated before running any experiment, meanwhile the second one based on dynamic topics was generated for every input concept during experimentation (see Section 4.2.2).

To describe the search-by-concept process carried out with the distributional thesaurus semantic spaces, an example for one concept in both spaces is explained:

```
Input concept: bone, pain, crack

1.TSSI

bone -> 0, 0.06360, 0.08038, 0, 0.09414, 0, 0, 0, 0, 0, 0,
0, 0, 0.07781, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

crack -> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.05004, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0.05667, 0

pain -> 0, 0, 0, 0, 0, 0, 0, 0, 0.11681, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.08805, 0, 0

Average vector -> 0, 0.02120, 0.02679, 0, 0.03138, 0, 0, 0,
0.03893, 0, 0, 0, 0, 0.04261, 0, 0, 0, 0, 0, 0, 0, 0,
0.02935, 0.01889, 0
```

After the semantic space analysis using *t=7*, the dictionary displayed seven words ranked in descending order as shown in Table 5.4. Notice that the system output includes two numeric values. The first one refers to the product of the pairwise semantic similarity between each noun member of the input concept and the respective candidate word, the second one consists of the Euclidean distance between the average vector and the candidate word vector.

Table 5.4: Dictionary output for the input concept: bone, pain, crack. TSSI

Product of pairwise similarity	**Euclidean distance**	**Target word**
0.000651	0.071281	wound
0.000624	0.101771	injury
0.000491	0.100060	infection
0.000381	0.110086	strain
0.000372	0.154031	damage
0.000337	0.072895	scar
0.000288	0.099984	cancer

`2. TSSII`

In order to obtain the vectors of each noun member of the input concept, it was necessary to determine the dimensionality of the space through the dynamic topics. In this case:

n=3 ; k=8

So, for each noun member of the input, their eight most similar terms were extracted from the Thesarus DB being the following:

```
bone - fracture, skull, ligament, tissue, skeleton, skin,
muscle, tendon

pain - headache, discomfort, nausea, soreness, suffering,
fatigue, trauma, grief

crack - hole, cocaine, fissure, leak, defect, flaw, seam,
marijuana

Dynamic topics:
“marijuana, seam, flaw, defect, leak, fissure, cocaine, hole,
grief, trauma, fatigue, suffering, soreness, nausea, discomfort,
headache, tendon, muscle, skin, skeleton, tissue, ligament,
skull, fracture”
```

The joint of all similar terms represented the dimensionality of the semantic space, in this case 24 dimensions. Having that, word vectors could be calculated having dimension values consisting of the semantic similarity measured between the word being analyzed and each dynamic topic. With the semantic space complete, the reverse search started getting the vectors of each input noun.

```
bone -> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.09142, 0, 0, 0,
0.13507, 0.13679, 0.13902, 0.14112, 0.14145, 0.14711, 0.15804,
0.17439

pain -> 0, 0, 0, 0.07881, 0, 0, 0, 0, 0.15659, 0.16100, 0.16853,
0.16890, 0.16949, 0.17526, 0.18057, 0.18560, 0.07990, 0.11230,
0, 0, 0, 0.08021, 0, 0.11785

crack -> 0.07864, 0.08159, 0.08774, 0.08909, 0.09208, 0.09224,
0.09996, 0.10696, 0, 0.04803, 0.04711, 0, 0, 0, 0.04391,
0.04314, 0, 0, 0, 0, 0, 0, 0, 0.08228

Average vector ->
0.02621, 0.02719, 0.02924, 0.05597, 0.03069, 0.03074, 0.03332,
0.03565, 0.05219, 0.06968, 0.07188, 0.05630, 0.08697, 0.05842,
0.07482, 0.07624, 0.07165, 0.08303, 0.04634, 0.04704, 0.04715,
0.07577, 0.05268, 0.12484
```

Again, once the semantic space analysis was completed, the dictionary displayed the list of seven words shown in Table 5.5 ranked in descending order.

Table 5.4 and Table 5.5 show the results obtained from both semantic spaces created using the distributional thesaurus. Many observations were

Table 5.5: Dictionary output for the input concept: bone, pain, crack. TSSII

Product of pairwise similarity	**Euclidean distance**	**Target word**
0.001691	0.292264	fracture
0.000685	0.212339	bruise
0.000651	0.150066	wound
0.000624	0.223406	injury
0.000491	0.255811	infection
0.000478	0.196695	swelling
0.000381	0.165560	strain

obtained during the analysis of results. First of all, as opposed to WordNet, the output does not include the gloss of the target words. Although the word gloss is not part of the system output proposed in this work, the additional information might improve the output comprehension in some cases such as a language learner using this dictionary.

In the case of TSSI, the sparseness in word vectors is notable; the reason is the linguistic context defining the dimensionality of the space represented by WordNet top concepts. Unlike WordNet graph where every noun is interconnected with generic concepts ensuring the existence of a semantic similarity value between them; in a distributional thesaurus the semantic similarity depends on the features shared between words, thus, the existence of a feature between a generic concept (top concepts) and another word was not very common, having lots of zeros inside word vectors. Despite this handicap, the list of output words seemed to associate properly the semantic of the input nouns.

In the case of TSSII, the sparseness in word vectors decreased considerably due to the dynamic topics. Also, although depending on the same parameters as TSSI, it is appreciated the existence of new vocabulary in the list of output words. These new words associate properly the semantics of the input nouns, thus improving the quality of the dictionary output. The rest of the output words tend to be the same as in TSSI. The reason why the new vocabulary was missing in TSSI output was the non-existence of features between them and the WordNet top concepts, and words having a null vector were not considered as members of the semantic space.

Finally, for both semantic spaces, the relevance of the product of pairwise similarity between the target word and each input noun for ranking is observed. However, the determining factor in the target word selection is the existence of a pairwise similarity between the candidate word and each input noun. In this experiment, all target words had pairwise similarity defined with all input nouns in the Thesaurus DB. If target words with complete pairwise similarity did not fill the top-t output, then, a backoff was carried out until the output was complete. Another observation is the Euclidean distance of the target words whose values tend to be consistently low and not necessarily close with the threshold values proposed. The threshold values were determined empirically and their main purpose was to decrease the number of candidate words included in the reverse search by discarding those having no relationship with the input concept.

Experimentation with the thesaurus semantic spaces finished after implementing the complete test set. The reverse search of six different concepts with the three highest ranked output words using TSSI and TSSII are shown in Table 5.6. Notice how output words tend to be shared between them, but in some cases TSSII included new words that improved the results.

Table 5.6: Reverse search of six different concepts - Thesaurus Semantic Spaces

Concept	**System Results**	
	TSSI	**TSSII**
star, orbit, planet	*moon* *earth* *galaxy*	*moon* *earth* *mars*
hymn, flag, emblem	*logo* *banner* *slogan*	*logo* *national flag* *banner*
computing, data, science	*technology* *information technology* *research*	*technology* *information technology* *research*
tuning, string, tune, "musical instrument"	*instrument* *drum* *cloth*	*guitar* *instrument* *piano*
satellite, antenna, transmission	*equipment* *device* *radar*	*sensor* *transmitter* *equipment*
bacteria, disease, cold	*infection* *virus* *illness*	*infection* *virus* *illness*

After analyzing the results of the entire test set for both semantic spaces, it was concluded that the TSSI presented a vocabulary loss that affected directly the quality of the dictionary output, unlike TSSII where the vocabulary loss did not affect its output because the usage of dynamic topics ensured the permanence of any word related with at least one of the input nouns. Furthermore, in all the cases where new words were included in the dictionary output using TSSII, they were always improving the quality of the output. As a consequence, word vectors from TSSII were selected for the general evaluation.

5.1.3 Latent Dirichlet Allocation

LDA was the third source for semantic space construction proposed in this work. Words member of this semantic space had vectors of 100 elements representing 100 latent topics in Wikipedia corpus and the element values represent the probability of a word within each latent topic. Both the topics and the probabilities were determined automatically by LDA.

To describe the search-by-concept process carried out with LDA semantic spaces, an example for one concept is explained:

```
Input concept: classroom, student, professor

classroom ->
0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00007, 0.00627, 0.20007, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00001, 0, 0, 0, 0,
0.00066, 0, 0, 0, 0, 0, 0, 0.01138, 0, 0,  0.00002, 0.01841,
0.00126, 0, 0, 0, 0.00039, 0, 0.00001, 0.00005, 0, 0, 0, 0,
0.00004, 0.01449, 0.00001, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.00024, 0, 0, 0, 0.00122, 0.00143, 0.00580, 0, 0.00125, 0, 0,
0.00305, 0, 0.00056, 0, 0.00220, 0.01822, 0, 0, 0, 0, 0, 0,
0.00002, 0.00017, 0, 0, 0, 0

student ->
0, 0, 0, 0, 0, 0, 0.00547, 0, 0.00037, 0.00057, 0.00652,
0.04832, 0.14886, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00023,
0, 0, 0, 0, 0, 0, 0, 0.00857, 0, 0, 0.05893, 0.00536, 0, 0,
0.01517, 0, 0, 0.02925, 0.01021, 0.00003, 0, 0, 0.00806,
0.00676, 0, 0.00003, 0.01079, 0.00190, 0.00001, 0, 0.00808,
0.00081, 0.04549, 0, 0.00005, 0.00045, 0.00017, 0.02034,
0.00001, 0.00014, 0.00442, 0.00817, 0, 0.00005, 0.01385, 0, 0,
0.00939, 0.00134, 0.00005, 0.00002, 0, 0.00001, 0, 0, 0.00873,
0.00033, 0, 0.00437, 0, 0.00182, 0, 0, 0, 0.00931, 0.00346, 0,
0.00003, 0, 0, 0, 0, 0.00004

professor ->
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.02235, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0.00648, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00008,
0, 0, 0.00008, 0.00041, 0.00120, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0

Average vector ->
0, 0, 0, 0, 0, 0, 0.00182, 0, 0.00012, 0.00021, 0.00426,
0.08279, 0.05707, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.00007, 0, 0, 0, 0.000003, 0, 0, 0, 0, 0.00307, 0, 0,
0.01964, 0.00178, 0, 0, 0.01101, 0, 0, 0.00975, 0.00954,
0.00043, 0, 0, 0.00268, 0.00238, 0, 0.00001, 0.00361, 0.00063,
0.000003, 0.00002, 0.00269, 0.00028, 0.02002, 0.00014,
0.00041, 0.00015, 0.00005, 0.00678, 0.000003, 0.00004,
0.00147, 0.00272, 0, 0.00001, 0.00469, 0, 0, 0.00313, 0.00085,
0.00049, 0.00194, 0, 0.00042, 0, 0, 0.00392, 0.00011, 0.00018,
0.00145, 0.00073, 0.00668, 0, 0, 0, 0.00310, 0.00115, 0,
0.00001, 0.00005, 0, 0, 0, 0.00001
```

After the semantic space analysis using t=*7*, the dictionary displayed the seven words ranked in descending order shown in Table 5.7. Note that for LDA semantic space the unique parameter taken into account for output selection was the Euclidean distance of every word vector against the average vector of input nouns.

The complete test set was implemented for reverse search using LDA semantic space getting a significant varied level of semantic association; while for some input concepts their output words tend to be precise mixing input nouns meaning, for another part the quality of output words tend to decrease significatively. Table 5.8 shows the three highest ranked output words of six different concepts.

Once the test set was implemented with each source of semantic space, the

Table 5.7: Dictionary output for the input concept: classroom, student, profesor. LDA Semantic Space

Euclidean distance	**Target word**
0.046787	faculty
0.055337	graduate
0.057974	universities
0.067116	undergraduate
0.072096	courses
0.072369	university
0.079994	doctorate

Table 5.8: Reverse search of six different concepts - LDA Semantic Space

Concept	**System Results**
music, ballet, stage	musical performance composer
criminal, corruption, mafia	imprisonment murders trafficking
chain, battlefield, troop	trench resistant nutrients
cattle, barn, cow	buffalo herd rodeo
cake, balloon, candy	cigarette pie breakfast
filming, script, actor	productions theater comedy

experimentation phase of the search-by-concept dictionary was completed. All the results obtained during experimentation were used to initiate the evaluation, final part of this investigation.

5.2 Evaluation

One of the specific goals of this work is to determine which model of reverse search is closer to human associative reasoning; this section reports that task. It was necessary to find a way to measure the quality of the obtained results, so, complete results containing the three highest ranked output words of the proposed search-by-concept dictionary were reunited in one document. On the other hand, comparing against existing implementations was also important to define the significance of this work regarding the state of the art. There are two publicly accessible online sites that include systems allowing this kind of search: onelook.com and dictionary.com. Based on the state of the art [43], OneLook Reverse Dictionary was selected for comparison in terms of quality.

OneLook Reverse Dictionary lets the user describe a concept and get back a list of words and phrases related to that concept. This dictionary indexes hundreds of online dictionaries, encyclopedias, and other reference sites. Concerning the reverse lookup, it searches in their references for words that have definitions conceptually similar to the input concept [3].

Having the results from both sources, the search-by-concept dictionary and OneLook Reverse Dictionary, it was necessary to define the type of people in charge of the evaluation; Amazon Mechanical Turk [1] supported this requirement. Amazon Mechanical Turk is a marketplace for work that requires human intelligence by providing access to a virtual community of workers being able to choose from a variety of skill and capabilities of their workforce that fulfil the requester needs. 34 categorization master workers were selected to evaluate the quality of the results; the award of master qualification refers to a demonstration of accuracy in a type of Human Intelligence Task HIT, in this case categorization.

The HIT developed to evaluate the project results aimed to measure the association of semantics of output words taking into account the three highest ranked output of each source of reverse search including the results of the existing implementation. So, each input concept member of the test set became a numbered item and the results of the search-by-concept dictionary and OneLook Reverse Dictionary were listed below in four different sets. The evaluators needed to mark the degree of semantic association of each set of words with respect the input concept. The marks go from zero to three, meaning:

- 0 – no semantic association
- 1 – weak semantic association
- 2 – medium semantic association
- 3 – strong semantic association

Complete results shown in Tables 5.9 5.10 5.11 5.12 5.13 5.14 give an idea of how the HIT created on Amazon Mechanical Turk looked like, with the difference that the HIT did not had a specific order of output words sets; the sets of words position was random for every input concept in order to avoid a possible detection of the best source during evaluation that could led to a bias during evaluation.

Once we had the evaluation data of the 34 workers from Amazon Mechanical Turk, it was processed according to the project needs. First, the performance of each reverse search source with every input concept was determined by computing the mean of the marks values given by the workers (marks going from zero to three depending the degree of semantic association) as shown in Table 5.15. With this information, a comparison between the different sources of semantic space and the existing reverse dictionary was possible. So, for each input concept a comparison between mean values was carried out looking for the source with the highest one. Table 5.16 shows the results of this comparison noticing that both WordNet and the distributional thesaurus had a higher percentage of output words with better degree of semantic association comparing with Onelook Reverse Dictionary (RD), while LDA was the unique source of semantic space below the performance of OneLook RD.

To appreciate the distribution of semantic association degrees between the different sources evaluated in this work, the mean values of Table 5.15 were associated to a specific degree under the following consideration:

- Degrees of semantic association go from zero to three, being zero null association and three strong association. So, all values having a decimal part greater than or equal to .5 were rounded to their nearest unit.

Table 5.9: Complete Results. Part 1

Source \ Concept	1. genius poetry “love affair”	2. music dancing stage	3. chain battlefield troop
WordNet	romance charge musset	stage dancing musical composition section	ligament attachment wire
Distributional Thesaurus	sexual relationship affair philosopher	theater concert movie	operation force fighting
LDA	love passion fantasy	musical performances composer	trench resistant nutrients
OneLook Reverse Dictionary	novelist essays critic	choreography choreograph score	ch-47 chinook rage band
	4. criminal corruption mafia	5. glory flag battlefield	6. dark talk detective
WordNet	illumination wrongdoer bad person	emblem design marking	illumination semidarkness conversation
Distributional Thesaurus	violence crime fraud	rivalry success triumph	official government crisis
LDA	imprisonment murders trafficking	elaborate decorative decoration	encounter strange rings
OneLook Reverse Dictionary	mafioso gangster mob	stars and stripes union salute	aha shadow dirk gently
	7. motor wheel driver	8. nature evolution life	9. intelligence technology profession
WordNet	engine machine mechanical device	growth chemical reaction pressure	education teaching application
Distributional Thesaurus	car vehicle truck	history culture relationship	skill science education
LDA	pole laps fastest	mind sense ways	tool integrated technical
OneLook Reverse Dictionary	steering wheel brougham automobile	natural Huxley degenerate	arda extropy communication

Table 5.10: Complete Results. Part 2

Source \ Concept	10. classroom student professor	11. swimsuit hotel sand	12. alcohol cigarette drug
WordNet	enrollee academician educator	swimming trunks hostel overgarment	drug of abuse liquor whiskey
Distributional Thesaurus	school child pupil	beach restaurant apartment	cocaine tobacco marijuana
LDA	faculty graduate universities	resort mansion neighborhood	trials psyquiatric treat
OneLook Reverse Dictionary	monitor hall pass intern	morris lapidus maillot bathing trunks	straight high exhaust
	13. blood punch sport	14. cake balloon candy	15. stadium grass player
WordNet	athletic game outdoor game field game	cookie taffy waffle	gramineous plant herb ballplayer
Distributional Thesaurus	stuff food life	flower champagne ice cream	team game fan
LDA	infection breast liver	cigarette pie breakfast	wins draw scoring
OneLook Reverse Dictionary	foul boxing tampon	ice pound cake sponge cage	field lockhart stadium sod
	16. antenna screen broadcast	17. wheel motor “steering wheel”	18. “gym shoe” “athletic contest” race
WordNet	serial wide screen news program	handwheel machine mechanical device	sports meeting Olympic Games horse race
Distributional Thesaurus	camera radio cable	tire car vehicle	team sport candidate
LDA	ratings aerings viewers	torque exhaust brake	Houston driver chosen
OneLook Reverse Dictionary	set-top box tv-antenna television antenna	shimmy kingpin gear	pentathlon triathlon decathlon

Table 5.11: Complete Results. Part 3

Source \ Concept	19. cement rod sand	20. string tune tuning “musical instrument”	21. computer noise security
WordNet	pole mast spar	stringed instrument cord percussion instrument	sound safety boom
Distributional Thesaurus	steel concrete metal	guitar instrument piano	communication telecommunication technology
LDA	shallow cracks freezing	ensembles chord saxophone	mobile files database
OneLook Reverse Dictionary	mortar concrete aggregate	mandolin mandola violin	quiet hat hold
	22. candle priest christian	23. furniture door window	24. recipe ingredient oven
WordNet	Holy Order religious person Catholic	furnishing movable barrier framework	kitchen appliance broiler Dutch oven
Distributional Thesaurus	Catholic clergy Jew	wall chair bed	flavor food mixture
LDA	consecrated seminary clergy	standing foot sitting	baked boiled pultry
OneLook Reverse Dictionary	deacon padre druid	splay pane case	bake madeira cake dutch oven
	25. flour oven fire	26. bomb weapon battle	27. hair mirror comb
WordNet	combustión oxidation kitchen appliance	explosive device gun firearm	hairdo beard skin
Distributional Thesaurus	wáter salt brown sugar	attack violence forcé	glass bruise cloth
LDA	rifle bomb fired	combat enemy captured	shoes silk breeds
OneLook Reverse Dictionary	toast cake slow	mine charge broadax broadaxe	part matilla tease

Table 5.12: Complete Results. Part 4

Source \ Concept	28. bag clothing travel	29. bed siringe nurse	30. cadáver coffin crying sadness
WordNet	baggage garment case	bedroom furniture hypodermic syringe medical instrument	box reaction consumption
Distributional Thesaurus	food ítem goods	hospital furniture care	casket grief anger
LDA	dressed shoulder knife	drops pulls sleeping	shouts downstairs couch
OneLook Reverse Dictionary	weekender overnighter suitcase	shot inject bedside	bier facial expression weep
	31. wood nail “power saw”	32. costume dancing mask ballerina	33. star orbit planet
WordNet	power tool brad holdfast	disguise attire ballet dancer	sun superior planet satellite
Distributional Thesaurus	plastic metal brick	dance music cloth	moon earth mars
LDA	decorative furniture fountain	nude cowell selena	asteroid launch solar
OneLook Reverse Dictionary	sawmill scroll saw jigsaw	tutu carmagnole morris dance	satellite moon year
	34. thuderbolt cloud water	35. satellite antenna transmission	36. church pope Rome
WordNet	lightning atmospheric electricity precipitation	sputnik dissemination circulation	place of worship Catholic Christian
Distributional Thesaurus	gas metal electricity	sensor transmitter equipment	faith religion culture
LDA	gas materials carbon	planetary observatory galaxies	bishop priest chapel
OneLook Reverse Dictionary	thunder thundercloud rain	satellite dish long line (telecom) feed	uniate vatican holy see

Table 5.13: Complete Results. Part 5

	37. clown laugh show	38. car building people	39. bacteria disease cold
WordNet	performance comedian contest	motor vehicle self-propelled vehicle wheeled vehicle	eubacteria communicable disease microorganism
Distributional Thesaurus	drama comedy song	area school business	infection virus illness
LDA	celebrity designer talent	houses center open	syndrome infection brain
OneLook Reverse Dictionary	nye amuse carnival	house club prison	pneumonia infect bubonic plague
	40. cattle barn cow	41. field ball stick sport	42. sight smell taste
WordNet	bovine farm building beef	game equipment athletic game outdoor game	sensation perception aroma
Distributional Thesaurus	Livestock crop fish	game team life	flavor sound color
LDA	buffalo herd rodeo	tied doublé chess	breakfast cane frozen
OneLook Reverse Dictionary	calf cowshed cowbarn	lacrosse hockey bandy	nasty sense head
	43. brother grandparent cousin	44. filming script actor	45.hymn flag emblem
WordNet	male sibling forebear kinsman	photography pictorial representation performer	symbol religious song religious music
Distributional Thesaurus	father husband son	performance actress screenplay	logo national flag banner
LDA	younger uncle fathers	productions theater comedy	decorative elaborate furniture
OneLook Reverse Dictionary	german layzie bone lucius antonius	milos forman dave thompson jiri menzel	standard hammer and sickle agnus dei

Table 5.14: Complete Results. Part 6

	46. foam soap water clothing	47. bone pain crack	48. discipline map earth
WordNet	cleansing agent formulation salt	connective tissue animal tissue tissue	Jovian planet Jupiter ion
Distributional Thesaurus	cloth chemical paint	fracture bruise wound	structure guideline technique
LDA	gas fuel materials	infections liver lung	wisdom forth sphere
OneLook Reverse Dictionary	wash suds rinse	fracture twinge smart	survey geology globe

	49.birthday happiness surprise	50. computer data science
WordNet	nirvana (Hinduism) blessedness wonder	natural science humanistic discipline physics
Distributional Thesaurus	disappointment joy sadness	technology information software
LDA	desperate surprised furious	users code technology
OneLook Reverse Dictionary	glory heigh-ho millennium	input format buffer

Table 5.15: Semantic association mean values of the test set concepts

Concept \ Source	WordNet	DT	LDA	OneLook RD
1	1.74	2.29	0.88	1.26
2	2.35	1.21	2.5	2.38
3	0.53	2.18	0.97	0.74
4	1.53	2.41	2.15	2.68
5	1.21	1.85	0.35	2.06
6	1.44	0.5	0.59	0.94
7	1.97	2.18	1.35	1.74
8	1.03	1.21	0.59	0.79
9	1.62	2.12	1.44	0.29
10	2.47	1.94	2.47	1.41
11	1.82	1.35	1.21	1.12
12	2.44	2.5	0.47	0.91
13	1.35	0.21	0.24	1.68
14	1.26	0.88	0.26	1.41
15	1	1.85	1.18	2.09
16	1.59	1.71	1.62	2.29
17	1.68	2.03	1.26	0.71
18	1.47	1.74	0.38	2.12
19	0.59	2	0.44	1.82
20	1.85	2.15	1.44	2.29
21	1.24	1.68	1.53	0.26
22	2.29	1.82	2.03	1.38
23	1.56	1.97	0.35	0.62
24	1.53	1.56	1.41	2.12
25	1.32	0.85	0.29	1.41
26	2.62	1.79	1.71	1.41
27	1.88	0.24	0.44	0.97
28	2.18	0.41	0.47	1.97
29	2.03	1.88	0.79	2.12
30	0.53	2.18	0.24	1.32
31	1.62	0.68	0.21	1.85
32	2.32	1.88	0.29	1.47
33	2.03	2.24	1.71	1.68
34	2.5	0.82	0.47	2.59
35	0.88	1.62	0.94	2.21
36	2.24	1.74	2.12	1.88
37	2.03	1.44	0.71	1.5
38	1.03	1	0.97	0.97
39	1.94	2.41	1.44	1.76
40	2.15	1.18	1.32	2.44
41	2	1.38	0.24	1.79
42	2.35	2.15	0.47	1.12
43	1.68	2.24	1.68	0.18
44	1.79	2.41	1.82	0.56
45	1.68	1.94	0.76	0.74
46	1.59	1.12	0.29	2.29
47	0.88	2.32	0.38	1.91
48	0.65	0.79	1	1.82
49	0.68	0.85	0.71	0.68
50	1.21	2.29	1.94	1.32

Table 5.16: Comparison between sources of reverse search

Comparison	Best answer percentage
WN vs. OnelookRD	52.941 % vs 47.058%
DT vs. OnelookRD	52% vs 48%
LDA vs. Onelook RD	31.372% vs 68.627%

The distribution is resumed in Figure 5.1 representing a graphic that indicates, for the four sources of reverse search, the amount of output words belonging with each degree of semantic association.

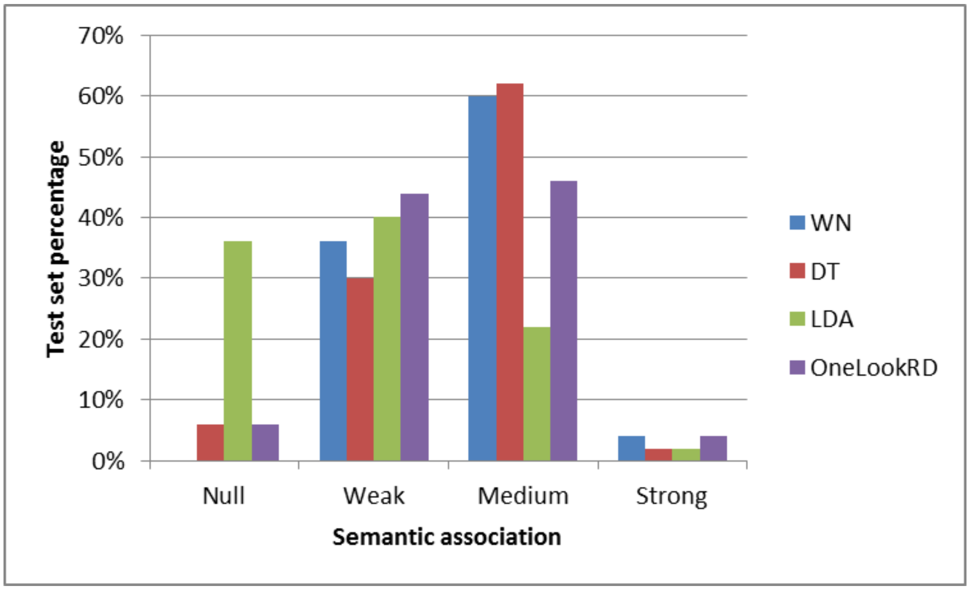


Figure 5.1: Semantic Association Distribution

The information exposed in Figure 5.1 indicates that the weak and medium degrees of semantic association concentrate most of the test set results. The strong degree of semantic association was barely achieved by the different sources of reverse search, WordNet and OneLook RD got a 4% while the distributional thesaurus and LDA got a 2% of the results. For the medium degree of semantic association, WordNet and the distributional thesaurus covered more than 50% of the output words reunited from the test set, with 60% for WordNet and 62% for the distributional thesaurus; meanwhile OneLook and LDA got a 46% and 22% respectively. A weak semantic association was achieved in 36% of WordNet output words, 30% for the distributional thesaurus and 44% for OneLook RD. LDA semantic space was the source of reverse search with the lowest performance having a 44% of weak semantic association and a 34% of null semantic association, covering almost the 80% of the test set output words within the poorest levels of association. The distributional thesaurus and OneLook RD got in 6% of the test set output words with null semantic association, being WordNet the unique source with no output words with null semantic association.

Finally, the overall performance for each source of reverse search was computed taking into account all mean values shown in Table 5.15. This calculation accomplished the ultimate specific goal proposed in this work, to determine which source of reverse search is closer to human associative reasoning; so, Table 5.17 shows the overall degree of semantic association achieved by each source of reverse search.

Table 5.17: Overall degree of semantic association

Source	**Overall degree of semantic association**
WordNet	1.627058824
DT	1.623529412
LDA	1.010588235
OneLook RD	1.501764706

According to Table 5.17 the best source of reverse search is WordNet concluding it is the closer to human associative reasoning. The overall performance value indicates a semantic association proximate to a medium degree, being also the case of the distributional thesaurus and OneLook RD, both of them under WordNet performance as the second and third best performance respectively. LDA overall performance indicates a semantic association barely over a weak degree, being the source of reverse search with the lowest results.

The analysis of the evaluation data ended after concluding the best source of reverse search; however, an additional analysis of the workers data was carried out in order to detect possible senseless evaluations that could have affected the performance of any source of reverse search.

The test set proposed for evaluation consisted of 50 concepts and each concept had four different sets of output words marked by the Amazon workers from zero to three depending on the degree of semantic association. The mean of the marks was calculated, ending with a total of 200 mean values. For every worker, the sum of the deviations of their marks from their respective mean value was computed. Then, the highest value of sums was identified and used to normalize the sum of deviations. Finally, workers with a normalization value above 0.78 were discarded in order to carry out a new analysis of the evaluation data. The marks given by these workers had a notable deviation from the mean throughout their test set evaluation, so, a new analysis without these workers might change the interpretations previously obtained. Table 5.18 shows the sum of deviation of the 34 workers and their normalization values.

As shown in Table 5.18, four workers had normalization values above 0.78. So, the new analysis of evaluation data would followed the same route as before but considering only the information of 30 workers. Again, the mean of the marks values given by the workers was computed for every input concept. These values are displayed in Table 5.19 and were used to carry out the comparison between the different sources of semantic space and OneLook RD. Table 5.20 shows the results of this comparison noticing that both WordNet and the distributional thesaurus had a better performance comparing with Onelook RD; moreover, WordNet best answer percentage increased almost a 4% from the original evaluation. Once more, LDA was the unique source of semantic space below the performance of OneLook RD.

To appreciate the new distribution of semantic association degrees, the graphic represented by Figure 5.2 shows the variations in the sources of reverse search after removing the data of four workers. The most significant changes are: WordNet reverse search now presents a 2% of output words with null semantic association, the distributional thesaurus increased its percentage of null semantic association in 4% while LDA decreased it in 2%; OneLook RD percentage kept it in 6%. On the other hand, the strong degree of semantic association increased in 2% for OneLook RD, while for the other sources remained unchanged. This made OneLook RD the source with the highest percentage of output words (a 6%) with strong semantic association; however, it is also the source with the highest percentage of output words having a weak semantic association as indicated by its 46%.

Despite the variations, the overall performances of the reverse search sources did not present considerable changes. As shown in Table 5.21, WordNet remains as the best source of reverse search according to the Amazon

Table 5.18: Deviations of Amazon Mechanical Turk workers

Worker	**Total deviation**	**Normalization value**
1	99.5588	0.5396
2	108.4411	0.5877
3	82.5588	0.4474
4	97.0882	0.5262
5	88.9705	0.4822
6	144.0294	0.7806
7	90.7352	0.4917
8	110.8529	0.6008
9	87.9705	0.4768
10	99.8529	0.5412
11	95.5588	0.5179
12	147.6176	0.8
13	114.6764	0.6215
14	108.7352	0.5893
15	114.0882	0.6183
16	115.3823	0.6253
17	143.7941	0.7793
18	99.5588	0.5396
19	96.2647	0.5217
20	94.3823	0.5115
21	121.147	0.6566
22	76.6764	0.4155
23	132.0294	0.7156
24	122.7941	0.6655
25	151.0882	0.8189
26	122.147	0.662
27	128.9117	0.6987
28	102.0294	0.553
29	122.5588	0.6642
30	122.0294	0.6614
31	117.0882	0.6346
32	113.0294	0.6162
33	184.5	1
34	85.9705	0.4659

Table 5.19: Semantic association mean values of the test set concepts. New analysis of evaluation data

Concept \ Source	WordNet	DT	LDA	OneLook RD
1	1.73	2.26	0.83	1.2
2	2.36	1.16	2.56	2.36
3	0.43	2.06	0.9	0.76
4	1.46	2.4	2.1	2.7
5	1.16	1.96	0.33	2.06
6	1.4	0.46	0.53	0.8
7	2	2.16	1.33	1.7
8	0.96	1.13	0.53	0.76
9	1.5	2.1	1.4	0.26
10	2.43	1.96	2.46	1.36
11	1.7	1.23	1.06	0.96
12	2.4	2.5	0.5	0.86
13	1.23	0.16	0.13	1.6
14	1.23	0.9	0.26	1.33
15	0.96	1.83	1.13	2.06
16	1.5	1.7	1.73	2.36
17	1.66	1.96	1.33	0.73
18	1.36	1.66	0.26	2.13
19	0.5	1.96	0.4	1.8
20	1.83	2.13	1.4	2.23
21	1.1	1.6	1.43	0.23
22	2.23	1.76	2	1.23
23	1.66	1.9	0.3	0.53
24	1.53	1.5	1.33	2.03
25	1.37	0.83	0.23	1.33
26	2.63	1.77	1.73	1.27
27	1.83	0.27	0.4	0.97
28	2.1	0.37	0.47	1.87
29	1.97	1.87	0.73	2.07
30	0.5	2.23	0.23	1.23
31	1.53	0.67	0.2	1.83
32	2.23	1.93	0.27	1.47
33	2,03	2.17	1.73	1.63
34	2.5	0.77	0.43	2.6
35	0.97	1.73	0.93	2.2
36	2.27	1.73	2.1	1.83
37	2.1	1.4	0.67	1.43
38	0.97	1	0.9	0.93
39	1.93	2.4	1.37	1.77
40	2.27	1.13	1.27	2.5
41	2	1.27	0.27	1.73
42	2.33	2.17	0.4	1.07
43	1.8	2.17	1.63	0.1
44	1.83	2.47	1.83	0.53
45	1.8	1.93	0.67	0.77
46	1.53	1.1	0.3	2.3
47	0.97	2.37	0.33	1.9
48	0.67	0.7	0.93	1.77
49	0.7	0.8	0.7	0.6
50	1.07	2.4	1.87	1.27

Table 5.20: Comparison between sources of reverse search. New analysis of evaluation data

Sources	Best answer percentage
WN vs. OnelookRD	56.862% vs 43.137%
DT vs. OnelookRD	52% vs 48%
LDA vs. Onelook RD	30% vs 70%

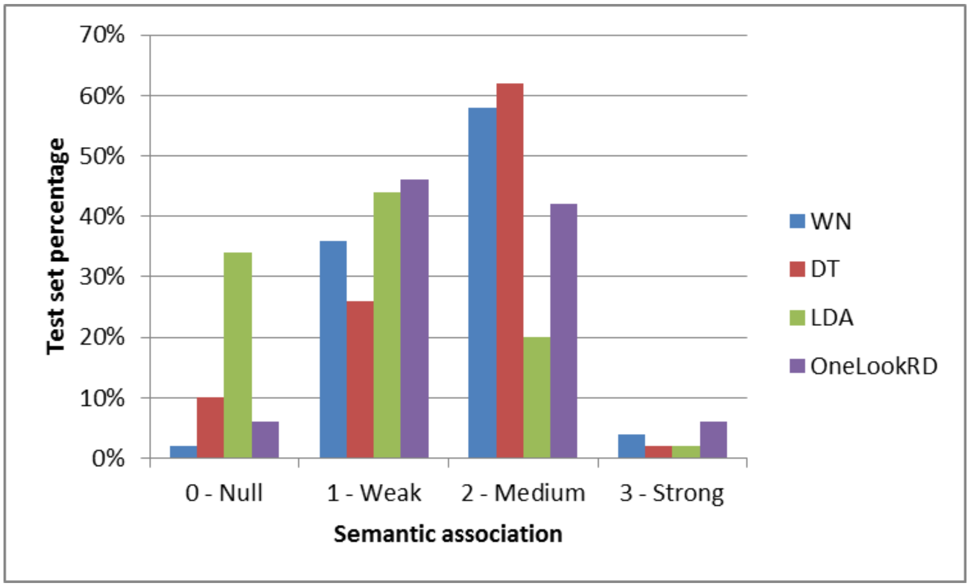


Figure 5.2: Semantic Association Distribution. New analysis of evaluation data

workers, followed by the distributional thesaurus, OneLook RD and LDA in that descending order. WordNet and the distributional thesaurus still have an overall semantic association proximate to a medium degree, meanwhile OneLook RD was closer to a weak degree of semantic association. LDA maintained a weak semantic association degree.

The poor performance of LDA semantic space derived on an additional revision to determine the possible causes. Word vectors of this semantic space had 100 elements, each one representing the word probability with a latent topic automatically discovered by LDA. However, despite the multiple di-

Table 5.21: Overall degree of semantic association. New analysis of evaluation data

Source	Overall degree of semantic association
WordNet	1.606
DT	1.603
LDA	0.978
OneLook RD	1.462

mensions, most of their values were zeros ending with sparse vectors. This suggested that most of the untagged topics generated by LDA might not be suitable representing a word feature, or at least not for the words forming the proposed test set. After reviewing the content of the topics, many of them were related to countries and states; also, it was noticed a significant presence of topics related with religion. This linguistic context is not useful representing the semantic of a word, thus, vector elements represented by these topics were filled with zeros in most of the cases. So, being LDA an unsupervised algorithm, this event was difficult to avoid unless directly modifying the algorithm structure.

5.3 Case analysis

The evaluation indicates that WordNet semantic space is the source of reverse search with the highest degree of semantic association after evaluating with the test set developed in this thesis, (see Table 5.21). However, it was important to carry out an analysis over the test set concepts and the sources of reverse search in order to detect any possible correlation.

One of the aspects to study was the distribution of the output words marked with the highest degree of semantic association and the subject involving their respective input concept in order to detect if there were cases of a source of reverse search giving the best marked output words for a specific subject. So far, we have only considered the overall degree of semantic association; this section shows the distribution of the best marked output words within sources of reverse search.

Another important aspect for consideration was the agreement between users during the evaluation process. For every concept, the variance between marks given by the users was calculated. Table 5.22 shows the variance values for each input concept and its respective source of reverse search, values in bold indicate the source evaluated with the highest degree of semantic association.

Analyzing Table 5.22 it is visible that the distribution of best answers is random and lies between WordNet, the Distributional Thesaurus and OneLook RD, being LDA semantic space the source of reverse search with the lowest number of best answers having only two. Based on this appreciation, the performance of the search-by-concept dictionary could be enhanced by implementing a blending method for WordNet and the Distributional Thesaurus, being able to display the most relevant output words depending on the input concept. This proposal needs further analysis in order to determine in which cases a source of reverse search obtains better results than other, so it is considered as part of the future work to improve further the investigation results.

On the other hand, the variance calculated for every input concept shows that the agreement between users was irregular having a wide variety of values, most of them indicating a medium variance throughout the test set. This does not reflect an incorrect evaluation process; on the contrary, it shows the complexity of human associative reasoning and the difficulties to evaluate it. Also, a correlation is not visible between low variance and the best answer's selection. In some cases the source of reverse search marked with the best degree of semantic association presents high variance, while in other cases the variance decreases. This occurs with all the sources of reverse search.

Accordingly, this work presented a new method for reverse search through word vector creation based on three different sources: WordNet, a distributional thesaurus and LDA. Two of the proposed methods performed better than an existing implementation: OneLook RD; being WordNet the best source for reverse lookup of the search-by-concept dictionary created during this investigation.

Table 5.22: Variance between users' marks

Concept \ Variance	WordNet	DT	LDA	OneLook RD
1. genius poetry "love affair"	0.328	**0.528**	0.405	0.226
2. music dancing stage	0.365	0.538	**0.378**	0.432
3. chain battlefield troop	0.312	**0.528**	0.490	0.445
4. criminal corruption mafia	0.648	0.373	0.490	**0.210**
5. glory flag battlefield	0.272	0.165	0.288	**0.062**
6. dark talk detective	**0.440**	0.448	0.448	0.493
7. motor wheel driver	0.333	**0.605**	0.622	0.543
8. nature evolution life	0.432	**0.582**	0.382	0.245
9. intelligence technology profession	0.383	**0.290**	0.506	0.262
10. classroom student professor	0.378	0.632	**0.382**	0.498
11. swimsuit hotel sand	**0.410**	0.378	0.262	0.432
12. alcohol cigarette drug	0.306	**0.383**	0.450	0.648
13. blood punch sport	0.378	0.205	0.182	**0.573**
14. cake balloon candy	0.378	0.490	0.195	**0.555**
15. stadium grass player	0.432	0.472	0.582	**0.528**
16. antenna screen broadcast	0.650	0.476	0.595	**0.432**
17. wheel motor "steering wheel"	0.622	**0.698**	0.688	0.595
18. "gym show" "athletic contest" race	0.232	0.355	0.262	**0.648**
19. cement rod sand	0.250	**0.565**	0.440	0.626
20. string tune tuning "musical instrument"	0.405	0.315	0.440	**0.445**
21. computer noise security	0.423	**0.440**	0.512	0.178
22. candle priest christian	**0.445**	0.378	0.466	0.578
23. furniture door window	0.488	**0.623**	0.276	0.315
24. recipe ingredient oven	0.648	0.583	0.622	**0.498**
25. flour oven fire	**0.498**	0.538	0.178	0.422
26. bomb weapon battle	**0.232**	0.312	0.528	0.595
27. hair mirror comb	**0.338**	0.195	0.373	0.765
28. bag clothing travel	**0.490**	0.232	0.248	0.915
29. bed syringe nurse	0.365	0.515	0.328	**0.662**
30. cadaver coffin crying sadness	0.316	**0.378**	0.312	0.445
31. wood nail "power saw"	0.648	0.422	0.226	**0.472**
32. costume dancing mask ballerina	**0.445**	0.395	0.195	0.448
33. star orbit planet	0.365	**0.338**	0.395	0.565
34. thunderbolt cloud water	0.316	0.245	0.378	**0.306**
35. satellite antenna transmission	0.365	0.595	0.528	**0.360**
36. church pope Rome	**0.328**	0.728	0.356	0.538
37. clown laugh show	**0.290**	0.306	0.422	0.645
38. car building people	0.298	**0.466**	0.423	0.328
39. bacteria disease cold	0.595	**0.306**	0.365	0.578
40. cattle barn cow	0.528	0.315	0.662	**0.383**
41. field ball stick sport	**0.533**	0.195	0.195	0.595
42. sight smell taste	**0.355**	0.538	0.373	0.262
43. brother grandparent cousin	0.626	**0.605**	0.632	0.090
44. filming script actor	0.472	**0.248**	0.272	0.648
45. hymn flag emblem	0.893	**0.395**	0.488	0.645
46. foam soap water clothing	0.582	0.356	0.276	**0.343**
47. bone pain crack	0.498	**0.365**	0.222	0.690
48. discipline map earth	0.488	0.476	0.395	**0.512**
49. birthday happiness surprise	0.276	**0.426**	0.276	0.306
50. cumputer data science	0.528	**0.373**	0.582	0.728

Chapter 6

Conclusions

The search-by-concept dictionary developed in this work demonstrated a good performance after being tested with input concepts covering a wide range of subjects. The proposal consisted of a new method for reverse dictionary construction with a semantic approach. The semantic approach was achieved through the implementation of words as vectors; the semantic properties of a word were captured in the vector elements determined by a given linguistic context. Three sources were used for word vector construction: WordNet, a distributional thesaurus and LDA; each source constituted a Semantic Space. The search-by-concept dictionary showed, with two semantic spaces (WordNet and the distributional thesaurus), a better performance after having been compared with existing implementations.

The analysis of the evaluation data revealed that semantic similarity measures performed well used as source for word vectors creation. Word vectors from WordNet semantic space were created using the JCN semantic similarity measure, while word vectors from the distributional thesaurus semantic space were created using Lin semantic similarity measure. Also, distributed representation based on WordNet top concepts achieved good levels of generalization, capturing word semantic features properly.

On the other hand, LDA semantic space presented the poorest results. Probably, the automatic generation of topics was the cause, having low-informative topics generating zeros inside word vectors in most of the cases. Many topics related with countries, states and religion were detected. The evaluation data revealed that this linguistic context was not useful representing the semantic of a word.

Another aspect to highlight was the good performance of the semantic space analysis proposed for reverse search which in most cases seemed to have merged properly the characteristics of the words forming the input nouns. The main conclusion of this work is that vector space word representation gives promising results for reverse dictionary construction.

6.1 Future Work

Given a phrase describing a desired concept of idea, a reverse dictionary provides target words whose definitions match the entered phrase. In the search-by-concept dictionary developed in this work, input phrases considered only nouns. So, the first future work proposal is to allow input phrases including words of any part of speech.

WordNet semantic space implemented JCN semantic similarity measure for word vector creation. The evaluation data showed that WordNet was the best source of reverse search followed by the distributional thesaurus which included word vectors based on Lin's semantic similarity measure. Another future work proposal is to experiment with a distributional thesaurus based on JCN semantic similarity measure in order to improve its semantic space performance. Also, modify the parameters during the distributional thesaurus creation to increase the vocabulary. Words occurring at least 100 times in the corpus were considered, but it is possible to decrease the number of times and observe the impact in the vocabulary forming the thesaurus semantic space.

Taking into account the fact that WordNet and the distributional thesaurus semantic spaces had the highest overall degree of semantic association, and that the output words of both sources marked as the best answers conformed over 50% of the test set, as future work is proposed to experiment with blending methods in order to improve results by mixing two or more of the techniques explored in this investigation.

Regarding LDA semantic space, it could be implemented a semi-supervised LDA [7] to tag the latent topics in Wikipedia corpus in order to improve its semantic space. Experimentation could include a tagging based on WordNet top concepts due to the good performance observed in word vectors that used them as linguistic context. Also, adding a stemming phase during corpus processing aiming to generate more informative topics; LDA traditional implementations [6] do not include this phase, that is the reason why it was not considered for the first experiments.

Finally, it is possible to increase the size of the test set looking to extend the lexical scope, always seeking for an uniform distribution of subjects among it in order to guarantee an equilibrated test set.

6.2 Publications

During this research development, the following article was published (Scopus indexed):

- Méndez, O., Calvo, H., & Moreno-Armendáriz, M. A. (2013). A Reverse Dictionary Based on Semantic Analysis Using WordNet. In *Advances in Artificial Intelligence and its Applications* (pp. 275-285). Springer Berlin Heidelberg.

Appendix A

Publications

A Reverse Dictionary Based on Semantic Analysis Using WordNet

Oscar Méndez, Hiram Calvo, and Marco A. Moreno-Armendáriz

Centro de Investigación en Computación - Instituto Politécnico Nacional
Av. Juan de Dios Bátiz, 07738, Distrito Federal, México
omendez_a12@sagitario.cic.ipn.mx

Abstract. In this research we present a new approach for reverse dictionary creation, one purely semantic. We focus on a semantic analysis of input phrases using semantic similarity measures to represent words as vectors in a semantic space previously created assisted by WordNet. Then, applying algebraic analysis we select a sample of candidate words which passes through a filtering process and a ranking phase. Finally, a predefined number of output target words are displayed. A test set of 50 input concepts was created in order to evaluate our system, comparing our experimental results against OneLook Reverse Dictionary to demonstrate that our system provides better results over current available implementations.

Keywords: reverse dictionary, semantic analysis, search by concept, vector space model.

1 Introduction

Over the years, people have used dictionaries for two well-defined purposes. Both of them are reflected on the dictionary's definition that is a collection of words listed alphabetically in a specific language, which contains their usage informations, definitions, etymologies, phonetics, pronunciations, and other linguistic features; or a collection of words in one language with their equivalents in another, also known as a lexicon. When these different ideas come together we understand why this resource hasn't lost importance and continue to be widely used around the world.

As part of the technological evolution the world has experienced during the last years, dictionaries are now available in electronic format. This resource has different advantages over the traditional printed dictionary, being the most important the easy access that it allows users and the very fast response time. Lexicographers constantly improve this resource, in order to assist language users, by increasing the number of words defined in the dictionary and adding lots more information associated with each one of them. Its performance is simple, just mapping words to their definitions, i.e. it does a lookup based on the correct spelling of the input word to find the definition.

F. Castro, A. Gelbukh, and M. González (Eds.): MICAI 2013, Part I, LNAI 8265, pp. 275–285, 2013.

This traditional approach is really helpful mostly for readers and language students, but isn't good enough taking into account the perspective of people who produce language. We all have experienced the problem of being unable to express a word that represents an idea in our mind although we are conscious of related terms, a partial description, even the definition. This may be due to a lack of knowledge in the word's meaning or a recall problem. People mainly affected by this problem are writers, speakers, students, scientists, advertising professionals, among others. For them, traditional dictionary searches are often unsuccessful because these kind of search demands an exact input, while a language producer tends to require a reverse search where the input are a group of words forming a formal definition or just a series of related terms, and the output is a target word.

The need for a different search access mode in a dictionary led to the creation of a reverse dictionary. Its basic objective is to retrieve a target word when a group of words which appear in its definition are entered. In other words, given a phrase describing a desired concept or idea, the reverse dictionary provides words whose definitions match the entered phrase. The chances of giving an exact definition of a concept is very difficult so synonym words or related words could also be considered during the search.

In this research we developed a new method to generate a reverse dictionary based on a large lexical English database known as WordNet and the implementation of different semantic similarity measures which help us in the generation of a semantic space.

2 State of the Art

Only three printed reverse dictionaries exist for English language. The reason is probably the complexity of its elaboration, especially the fact of choosing the proper form to distribute the information. The Bernstein's Reverse Dictionary [4] was the first of its kind, in this book, the definitions of 13,390 words were reduced to their most brief form and then ordered alphabetically.

With the availability of dictionaries in electronic format, the interest for a reverse lookup application has been growing during the last years. Unlike printed versions, several attempts have been made in the creation of the reverse lookup method seeking for the best performance.

In the reverse electronic dictionary presented in [7], synonyms were used to expand search capabilities. They create a dictionary database with words numerically encoded for quick and easy access; adding also synonym group numeric codes in order to extend the searching process. In every search the numeric codes of the input words are found and stored. Then, main entry words having the numeric codes of the input words within their definitions are located and displayed as output candidates.

The magnitude of this natural language application is appreciated when dictionaries for different languages are constructed like [5]. For this Japanese reverse dictionary three different databases were created, using traditional IR concepts. Each database stored all dictionary words (EDR, 1995) with their definitions as vectors, reflecting the term frequencies in each definition, with standard similarity metrics values (tf-idf, tf, binary values) as its elements. The reverse lookup method is separated in two stages. First, they parse the input concept with a morphological analyzer and create its vector, and then compare to the definition vectors to obtain the closest matching concept in the dictionary. To calculate the similarity between vectors they used cosine measure.

A different reverse lookup method was created in [8]. Their algorithm for French language does a reverse search using two main mechanisms. The first one extracts sets of words, from their lexical database of French words, which delimit the search space. For example, in the definition ‘a person who sells food’ the algorithm extracts all the sets of persons. The second mechanism computes a semantic distance between each candidate word in the extracted sets and the input definition to rank the output words. This latter value is based on the distances in the semantic graph, generated by their database, between hypernyms and hyponyms of the words being analyzed.

Another proposal was based on the notion of association: every idea, concept or word is connected [14]. Given a concept (system input) and following the links (associations) between input members, a target word would be reached. They proposed a huge semantic network composed of nodes (words and concepts) and links (associations), with either being able to activate the other.

In [15] the reverse lookup method depends on an association matrix composed of target words and their access keys (definition elements, related concepts). Two different sources were selected as corpus for the databases: WordNet and Wikipedia. The one based on WordNet used as target words the words defined in the dictionary and as access keys their definitions. The corpus based on Wikipedia used the page’s raw text as target words (after a filtering process) and the words co-occurrences within a given window of specific size as access keys. Finally for every input phrase, their members are identified and the reverse search results in a list of words whose vectors contain the same input terms.

The most recent reverse dictionary application we found is shown in [12]. To construct their database they created for every relevant term t in the dictionary its Reverse Mapping Set (RMS) which requires finding all words in whose definition relevant term t appears. For every input phrase a stemming process is required, then a comparison is made between the input and the RMS looking for the words whose definitions contain the input members; this generates a group of candidates that pass through a ranking phase based on similarity values computed using a similarity measure implemented on WordNet and a parser.

The systems presented above share different methodological features. All of them consider not only the terms extracted from the user input phrase, but also terms similar or related to them (synonyms, hyponyms, hyperyms) and also needed a previous dictionary processing in order to form their databases. The reverse search done by [7] [14] [15] and [12] at some point of its procedure does a comparison between the user input phrase to every definition in their databases looking for definitions containing the same words as the user input phrase, while [8] and [5] based their reverse search on the highest similarity values measuring graph distances and cosine respectively. All of this demonstrates a tendency during reverse lookup algorithms creation until now.

Our proposal presents a new approach for reverse dictionary creation, one purely semantic. We focus on a semantic analysis of input phrases using semantic similarity measures to represent words as vectors in a semantic space previously created assisted by WordNet. Then, applying algebraic analysis we select a sample of candidate words which passes through a filtering process and a ranking phase. Finally, a predefined number of output target words are displayed. It's important to mention that this project considers only nouns as word members of the semantic space, this part of speech restriction is due to the form in which vectors are constructed and the fact that it's only possible to calculate semantic similarity or semantic relatedness with words that belong to the same part of speech. Besides, it is well known that in natural language, concepts are expressed mostly as noun phrases [13].

3 WordNet as a Resource for Semantic Analysis

WordNet is a large lexical database for English and other languages. It groups words into sets of synonyms called synsets and describes relations between them. Lexical relations hold between word forms and semantic relations hold between word meanings.

The structure of word strings in WordNet specifies a specific sense of a specific word as shown below; this is used to avoid word sense disambiguation problems:

word#pos#sense

where pos is the part of speech of the word and its sense is represented by an integer number.

WordNet has a hierarchical semantic organization of its words, also called by computer scientists as "inheritance system" because of the inherited information that specific items (hyponyms) get from their superordinates. There are two forms to construe the hierarchical principle. The first one considers all nouns are contained in a single hierarchy. The second one proposes the partition of the nouns with a set of semantic primes representing the most generic concepts and unique beginners of different hierarchies [11]. To create WordNet's semantic space this project makes use of the second form and 25 top concepts were defined as semantic primes to represent the dimensions of word vectors.

The top concepts, with its specific sense, that were chosen are:

activity#n#1, animal#n#1 artifact#n#1, attribute#n#2, body#n#1,
cognition#n#1, communication#n#2, event#n#1, feeling#n#1, food#n#1,
group#n#1, location#n#1, motive#n#1, natural_object#n#1,
natural_phenomenon#n#1, human_ being#n#1, plant#n#2, possession#n#2,
process#n#6, quantity#n#1, relation#n#1, shape#n#2, state#n#1,
substance#n#1, time#n#5

This is also the order given to the top concepts during the vector representation of words mentioned further on.

WordNet also includes the implementation of similarity and relatedness measures. A semantic relatedness measure uses all WordNet's relations for its calculation meanwhile a semantic similarity measure only uses the hyponymy relation. Three measures were considered for database construction: Jiang and Conrath (JCN) [9], Lin [10] and the Lesk algorithm (Lesk) [2]. The first two are similarity measures which have demonstrated to have a good performance among other measures that use WordNet as their knowledge source [6]; the last one is an adaptation of the original Lesk relatedness measure that take advantage of WordNet's resources [1].

Jiang and Conrath: this measure combines the edge-based notion with the information content approach. It calculates the conditional probability of encountering an instance of a child-synset given an instance of a parent synset, specifically their lowest super-ordinate (lso). The formula is expressed in 1.

$$dist_{JCN}(c_1, c_2) = 2\log(p(lso(c_1, c_2))) - (\log(p(c_1)) + \log(p(c_2))) \qquad (1)$$

Lin: based on his similarity theorem: "The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are." It uses the same elements of JCN measure but in a different way. The formula is expressed in 2.

$$sim_{LIN}(c_1, c_2) = \frac{2\log p(lso(c_1, c_2))}{\log p(c_1) + \log p(c_2)} \qquad (2)$$

Lesk: the original algorithm measures the relatedness between two words by the overlap between their corresponding definitions as provided by a dictionary. Basically the steps are:

1. Retrieve from an electronic dictionary all sense definitions of the words to be measured.
2. Determine the definition overlap for all possible sense combinations.
3. Choose senses that lead to highest overlap.

In WordNet an extended gloss overlap measure is available, which combines the advantages of gloss overlaps with the structure of a concept hierarchy to create an extended view of relatedness between synsets [1].

4 Semantic Space Construction

In this section we describe the construction process of the semantic space that contains the numeric representation of all WordNet's nouns as vectors of 25 dimensions determined by the top concepts mentioned before. For every noun we create its vector measuring semantic similarity between the word and each top concept, then it is stored in the semantic space. After reading and creating the vectors for every noun, the process ends. This procedure is detailed in Figure 1.

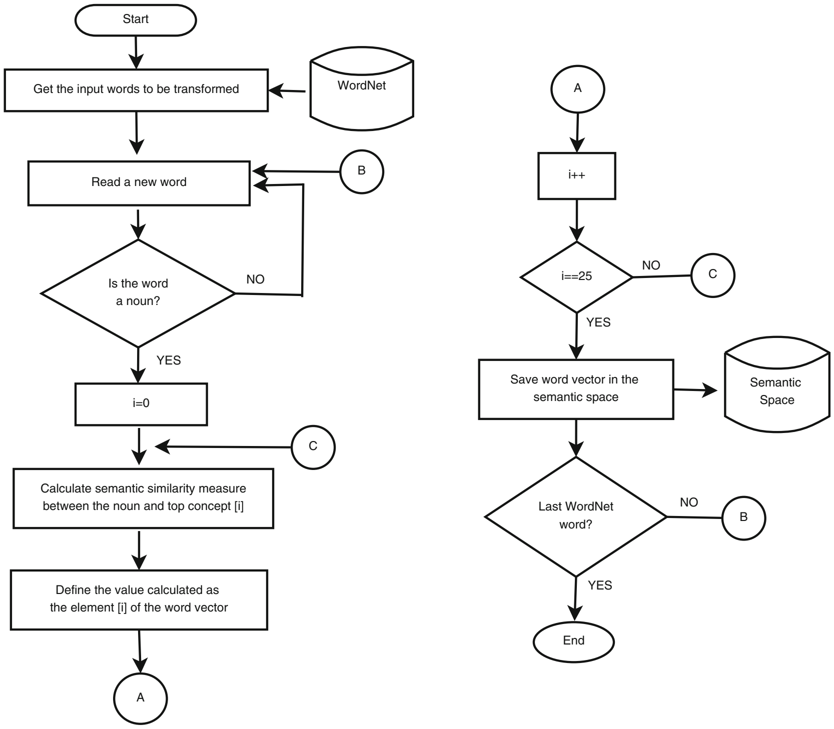

Start
Get the input words to be transformed
WordNet
Read a new word
B
Is the word a noun?
NO
YES
i=0
C
Calculate semantic similarity measure between the noun and top concept [i]
Define the value calculated as the element [i] of the word vector
A
A
i++
i==25
NO
C
YES
Save word vector in the semantic space
Semantic Space
Last WordNet word?
NO
B
YES
End

Fig. 1. Vector creation algorithm

The process was repeated for each of the different measures mentioned above, resulting on a semantic space with JCN measured vectors, another with Lin measured vectors, and the last one with Lesk measured vectors.

With the databases created, a normalization procedure was performed. For all word vectors maximum values of each dimensions were obtained in order to subsequently divide all word vectors dimensions by each respective maximum value previously obtained. Finally we have word vectors inside the semantic space with this form:

Genius → 0.05748, 0.04058, 0.09603, 0.06138, 0.06117, 0.04774, 0.07306, 0.02822, 0.06301, 0.07750, 0.05024, 0.05693, 0.03530, 0.12316, 0.01008, 0.01046, 0.00898, 0.05117, 0.03144, 0.05603, 0.04203, 0.07932, 0.03364, 0.02163, 0.07081

5 Search-by-Concept Process

A reverse dictionary receives a definition as input and gets a word that represents that concept as output. The search-by-concept dictionary proposed in this project is based on this principle.

The system input consists of a concept formed of n nouns. Once the input is defined, the system looks for the word vectors of their n components in the database and calculates their average. This gives as a result a new vector that should be located in the semantic space representing the word that combines all the characteristics given on the input concept. Regardless of whether the new vector already exists in the semantic space representing a word, a sample of twelve neighbor vectors is taken. This sample selection considers two parameters:

1. The euclidean distance value between vectors need to be:
   (a) For JCN less than 0.1
   (b) For Lin less than 0.8
   (c) For LSK less than 0.1

   These threshold values were determined after numerous testing. For vectors with euclidean distances bigger than the values mentioned above, the words they represented tend to have no relationship with the input concepts.
2. The product of the semantic similarity measure between each member of the input and the word represented by the neighbor vector is calculated; the top n words with the highest values are chosen to form the system output.

6 Results

Before showing some results, a complete example for JCN semantic space is shown below:

```
Input concept - gym_shoe#n#1 athletic_contest#n#1 race#n#2

gym_shoe#n#1 ->
0.05383, 0.03492, 0.11093, 0.05720, 0.05738, 0.04458, 0.06833, 0.02628,
0.05933, 0.07286, 0.04694, 0.05249, 0.03347, 0.11451, 0.00944, 0.00927,
0.00782, 0.04819, 0.02938, 0.05237, 0.03932, 0.07490, 0.03153, 0.02020,
0.06700

athletic_contest#n#1 ->
0.08950, 0.03136, 0.07729, 0.07229, 0.05214, 0.06832, 0.08528, 0.04630,
0.07227, 0.07297, 0.05879, 0.04805, 0.04601, 0.10280, 0.00946, 0.00849,
0.00708, 0.05869, 0.02943, 0.06550, 0.04902, 0.09036, 0.02934, 0.02281,
0.08023

race#n#2 ->
0.09333, 0.03214, 0.07960, 0.07480, 0.05331, 0.07113, 0.08805, 0.04859,
0.07433, 0.07472, 0.06073, 0.04942, 0.04732, 0.10539, 0.00970, 0.00866,
0.00724, 0.06035, 0.03020, 0.06766, 0.05061, 0.09279, 0.03003, 0.02350,
0.08229
```

```
Average vector ->
0.07888, 0.03280, 0.08927, 0.06809, 0.05427, 0.06134, 0.08055, 0.04039,
0.06864, 0.07351, 0.05548, 0.04998, 0.04226, 0.10756, 0.00953, 0.00880,
0.00738, 0.05574, 0.02967, 0.06184, 0.04631, 0.08601, 0.03030, 0.02217,
0.07650
```

After the search-by-concept process these are the results:

The seven output words with highest ranking are shown in Table 1. The most relevant result is `meet\#n\#1`. The proximity of its vector's dimensions values with the ones of the average vector previously calculated is notable.

```
meet#n#1 ->
0.08617, 0.03065, 0.07523, 0.07008, 0.05108, 0.06587, 0.08282, 0.04433,
0.07043, 0.07140, 0.05706, 0.04682, 0.04483, 0.10047, 0.00925, 0.00833,
0.00693, 0.05719, 0.02873, 0.06359, 0.04762, 0.08818, 0.02873, 0.02220,
0.07838
```

Table 1. System output for concept: gym_shoe#n#1 athletic_contest#n#1 race#n#2

Product of semantic similarity values	**Euclidean distance**	**Word**	**Gloss**
0.02642	0.02015	meet#n#1	a meeting at which a number of athletic contests are held
0.00580	0.02755	Olympic_Games#n#1	the modern revival of the ancient games held once every 4 years in a selected country
0.00426	0.02755	horse_race#n#1	a contest of speed between horses
0.00426	0.02755	footrace#n#1	a race run on foot
0.00387	0.05936	game#n#2	a single play of a sport or other contest
0.00325	0.03846	track_meet#n#1	a track and field competition between two or more teams
0.00293	0.04428	race#n#1	any competition

This process is done with the three different semantic spaces for every input concept. Table 2 and Table 3 show the reverse search of three different concepts with the two highest ranked output words from our system and the two highest ranked output words from an existing reverse dictionary [3] (OneLook Reverse Dictionary Online) respectively for comparison terms.

Table 2. Reverse search for three different concepts - System output

Concept	**System results**		
nature evolution life	JCN	growth#n#2	A progression from simpler to more complex forms.
		chemical_reaction#n#1	(Chemistry) a process in which one or more substances are changed into others.
	Lesk	oxidative_phosphorylation#n#1	An enzymatic process in cell metabolism that synthesizes ATP from ADP.
		blooming#n#1	The organic process of bearing flowers.
	Lin	growth#n#2	A progression from simpler to more complex forms.
		heat_sink#n#1	A metal conductor specially designed to conduct (and radiate) heat.
antenna screen broadcast	JCN	serial#n#1	A serialized set of programs.
		wide_screen#n#1	A projection screen that is much wider than it is high.
	Lesk	rerun#n#1	A program that is broadcast again.
		receiver#n#1	Set that receives radio or tv signals.
	Lin	electrical_device#n#1	A device that produces or is powered by electricity.
		surface#n#1	The outer boundary of an artifact or a material layer constituting or resembling such a boundary.
thunderbolt cloud water	JCN	atmospheric_electricity#n#1	Electrical discharges in the atmosphere.
		precipitation#n#3	The falling to earth of any form of water.
	Lesk	atmospheric_electricity#n#1	Electrical discharges in the atmosphere.
		cumulus#n#1	A globular cloud.
	Lin	atmospheric_electricity#n#1	Electrical discharges in the atmosphere.
		atmospheric_phenomenon#n#1	A physical phenomenon associated with the atmosphere.

Table 3. Reverse search for three different concepts - OneLook Reverse Dictionary output

Concept	**OneLook Reverse Dictionary results**
nature evolution life	natural Huxley
antenna screen broadcast	set-top box tv-antenna
thunderbolt cloud water	thunder cloud

Table 4. Evaluation

Output source	**Aspect 1**	**Aspect 2**	
Our system	94%	JCN	42%
		Lin	6%
		Lesk	32%
OneLook Reverse Dictionary	74%	20%	

At first sight, the results of our system seem to be correct answers for each concept, but in which way could we measure the quality of our results? We create a test set with 50 different concepts and for each concept we show the two highest ranked output words from our system and the two highest ranked output words from OneLook Reverse Dictionary, as in Table 2 and Table 3. A group of 10 people evaluated the test set under the following considerations:

1. Indicate if the output words converges with their associative reasoning.
2. Indicate which one of the sources gave the best results. And in case our system output was selected, specify the source of semantic space.

We resume the evaluation information in Table 4. Analyzing its content, it is clear that the performance of our system is better than OneLook Reverse Dictionary. Not only in the proximity with human associative reasoning capacity, it also gave the best results during the reverse search; where the concepts obtained from JCN semantic space demonstrate to combine better the characteristics of meaning of the input phrases.

7 Conclusions

In this paper, we described a new method for reverse dictionary construction with a semantic approach. We proposed the creation of three different semantic spaces, each one containing vectors created from different sources of semantic similarity measures. Also we described the different parts that constitute our reverse search together with an example. Our experimental results show that our system provides better results over current available implementations, including an improved system output providing also the gloss of every output word. This is very helpful in terms of evaluation because the user doesn′t have to waste time looking for a definition in order to verify the quality of the output.

As future work we propose the creation of two new semantic spaces based on different resources, a distributional thesaurus and latent Dirichlet allocation(LDA). A distributional thesaurus is a thesaurus generated automatically from a corpus by finding words which occur in similar contexts to each other. Meanwhile LDA is a generative probabilistic model for collections of discrete data. This enables an analysis of reverse search from different approaches to determine which one is the closest to human associative reasoning. A supervised approach (WordNet), semi-supervised approach (distributional thesaurus) and unsupervised approach (LDA).

Acknowledgements. The authors wish to thank to Instituto Politécnico Nacional (SIP-IPN grants 20130018 and 20130086, COFAA-IPN and PIFI-IPN) and the government of Mexico (SNI and CONACYT) for providing the necessary support to carry out this research work.

References

1. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)
2. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. IJCAI 3, 805–810 (2003)
3. Beeferman, D.: Onelook Reverse Dictionary (2013), `http://www.onelook.com/reverse-dictionary.shtml` (accessed January-2013)
4. Bernstein, T., Wagner, J.: Bernstein's reverse dictionary. Quadrangle/New York Times Book Co. (1975)
5. Bilac, S., Watanabe, W., Hashimoto, T., Tokunaga, T., Tanaka, H.: Dictionary search based on the target word description. In: Proc. of the Tenth Annual Meeting of The Association for NLP (NLP 2004), pp. 556–559 (2004)
6. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics 32(1), 13–47 (2006)
7. Crawford, V., Hollow, T., Crawford, J.: Reverse electronic dictionary using synonyms to expand search capabilities. Patent, 07 1997; US 5649221 (1997)
8. Dutoit, D., Nugues, P.: A lexical database and an algorithm to find words from definitions (2002)
9. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
10. Lin, D.: An information-theoretic definition of similarity. In: ICML, vol. 98, pp. 296–304 (1998)
11. Miller, G.A.: Nouns in wordnet: a lexical inheritance system. International Journal of Lexicography 3(4), 245–264 (1990)
12. Shaw, R., Datta, A., VanderMeer, D., Dutta, K.: Building a scalable database-driven reverse dictionary. IEEE Transactions on Knowledge and Data Engineering 25(3), 528–540 (2013)
13. Sowa, J.F.: Conceptual structures: Information processing in mind and machine (1984)
14. Zock, M., Bilac, S.: Word lookup on the basis of associations: from an idea to a roadmap. In: Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries, ElectricDict 2004, pp. 29–35. Association for Computational Linguistics, Stroudsburg (2004)
15. Zock, M., Schwab, D.: Lexical access based on underspecified input. In: Proceedings of the workshop on Cognitive Aspects of the Lexicon, pp. 9–17. Association for Computational Linguistics (2008)

Bibliography

[1] Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome, 2005. [Online; accessed December-2013].

[2] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 136–145, London, UK, UK, 2002. Springer-Verlag.

[3] D. Beeferman. Onelook Reverse Dictionary. http://www.onelook.com/reverse-dictionary.shtml, 2009. [Online; accessed January-2013].

[4] T. Bernstein and J. Wagner. *Bernstein's reverse dictionary.* Quadrangle/New York Times Book Co., 1975.

[5] S. Bilac, W. Watanabe, T. Hashimoto, T. Tokunaga, and H. Tanaka. Dictionary search based on the target word description. In *Proc. of the Tenth Annual Meeting of The Association for NLP (NLP2004)*, pages 556–559, 2004.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[7] S. Bodrunova, S. Koltsov, O. Koltsova, S. I. Nikolenko, and A. Shimorina. Interval semi-supervised lda: Classifying needles in a haystack. In F. Castro, A. F. Gelbukh, and M. González, editors, *MICAI (1)*, volume 8265 of *Lecture Notes in Computer Science*, pages 265–274. Springer, 2013.

[8] A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on*

WordNet and other lexical resources, second meeting of the North American chapter of the Association for Computational Linguistics, 2001.

[9] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, Mar. 2006.

[10] H. Calvo, E. Gelbukh, and A. Kilgarriff. Distributional thesaurus vs. wordnet: A comparison of backoff techniques for unsupervised pp attachment. In *Proceedings of Sixth International Conference on Intelligent Text Processing and Computational Linguistics: CICLING05*, pages 172–182, 2005.

[11] F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. In *Proceedings of EMNLP*, pages 109–117, 2001.

[12] V. Crawford, T. Hollow, and J. Crawford. Reverse electronic dictionary using synonyms to expand search capabilities. Patent, 07 1997. US 5649221.

[13] B. De Finetti. *Theory of probability: a critical introductory treatment.* Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1974.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[15] D. Dutoit and P. Nugues. A lexical database and an algorithm to find words from definitions. In *ECAI*, pages 450–454, 2002.

[16] D. J. Edmonds. *The Oxford reverse dictionary.* Oxford University Press, 2002.

[17] O. Ferret and M. Zock. Enhancing electronic dictionaries with an index based on associations. In N. Calzolari, C. Cardie, and P. Isabelle, editors, *ACL*. The Association for Computer Linguistics, 2006.

[18] J. R. Firth. A synopsis of linguistic theory 1930–55 (special volume of the philological society), 1957.

[19] B. A. Frigyik, A. Kapila, and M. R. Gupta. Introduction to the dirichlet distribution and related processes. Technical report, UWEE Technical Report, 2010.

[20] G. Furnas, T. Landauer, L. Gomez, and T. Dumais. Statistical semantics: Analysis of the Potential Performance of Keyword Information Systems. *Bell System Technical Journal*, 62(6):1753–1806, 1983.

[21] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

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

[23] J. E. Kahn, R. D. Association, et al. *Reader's Digest reverse dictionary.* Reader's Digest Association Limited, 1989.

[24] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[25] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[26] W. Lowe. Towards a theory of semantic space. In *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, pages 576–581. Lawrence Erlbaum Associates, 2001.

[27] R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 545–552, New York, NY, USA, 2005. ACM.

[28] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[29] Merriam-Webster Online. Merriam-Webster Online Dictionary, 2009.

[30] T. Mikolov, W. tau Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. The Association for Computational Linguistics, 2013.

[31] G. A. Miller. Nouns in WordNet: A Lexical Inheritance System. *International Journal of Lexicography*, 3(4):245–264, Dec. 1990.

[32] G. A. Miller. Wordnet: A lexical database for english. `http://wordnetweb.princeton.edu/perl/webwn`, 1995. [Online; accessed January-2013].

[33] O. Méndez, H. Calvo, and M. A. Moreno-Armendariz. A reverse dictionary based on semantic analysis using wordnet. In F. Castro, A. F. Gelbukh, and M. González, editors, *MICAI (1)*, volume 8265 of *Lecture Notes in Computer Science*, pages 275–285. Springer, 2013.

[34] D. Mochihashi. lda, a latent dirichlet allocation package. `http://chasen.org/~daiti-m/dist/lda/`, 2004. [Online; accessed August-2013].

[35] M. Porter. An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 40(3):211–218, 2006.

[36] R. Prasher and P. B. Mangla. *Library and Information Science: Parameters and Perspectives : Essays in Honour of Prof. P.B. Mangla*. Number v. 1 in Concepts in communication, informatics & librarianship. Concept Publishing Company, 1997.

[37] P. Resnik and E. Hardisty. Gibbs Sampling for the Uninitiated. Technical Report CS-TR-4956, UMIACS-TR-2010-04, LAMP-153, University of Maryland, Apr. 2010.

[38] S. M. Ross. *Introduction to Probability Models, Ninth Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.

[39] S. J. Russell, P. Norvig, J. F. Candy, J. M. Malik, and D. D. Edwards. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[40] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.

[41] H. Schütze and J. Pedersen. A vector model for syntagmatic and paradigmatic relatedness. In *Proc. of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113, Oxford, England, 1993.

[42] H. Schütze. Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123, Mar. 1998.

[43] R. Shaw, A. Datta, D. VanderMeer, and K. Dutta. Building a scalable database-driven reverse dictionary. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):528–540, 2013.

[44] G. Sierra and L. Hernández. A proposal for building the knowledge base of onomasiological dictionaries. *Journal of Cognitive Science*, 12(3):215–232, 2011.

[45] D. Soergel. *Indexing Languages and Thesauri: Construction and Maintenance.* Information Sciences Series. Melville Publishing Company, 1974.

[46] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley Publishing Company, Reading, MA, 1984.

[47] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, Jan. 2010.

[48] L. Wanner. *Lexical functions in lexicography and natural language processing.* Studies in language companion series. J. Benjamins, 1996.

[49] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[50] M. Zock and S. Bilac. Word lookup on the basis of associations: from an idea to a roadmap. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, ElectricDict '04, pages 29–35, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[51] M. Zock and D. Schwab. Lexical access based on underspecified input. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, COGALEX '08, pages 9–17, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

Glossary

AI

Artificial Intelligence. The science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence. 1

DB

Database. An organized collection of data. 10, 51, 56, 63, 66

HIT

Human Intelligence Task. A specific task created on Amazon Mechanical Turk, a virtual community of workers. 70, 71

IR

Information Retrieval. The activity of obtaining information resources relevant to an information need from a collection of information resources, and the part of information science, which studies these activity. 1–4, 10

JCN

Jiang and Conrath. The creators of a semantic similarity measure having their name. 30, 45, 54, 59–62, 89, 90

LAA

Least Asymmetric Ancestor. Let M and N be words. $LAA(M,N)$is the set of nodes that are common ancestors of both words, that are not member of the LCA set and where each member of the LAA set has at least one child, which is an ancestor of M and not an ancestor of N. 13

LCA

Least Common Ancestor. A concept in graph theory and computer science. Let T be a rooted tree with n nodes. The least common ancestor between two nodes v and w is defined as the lowest node in T that has both v and w as descendants. 12, 13, 17

LDA

Latent Dirichlet Allocation. A generative probabilistic model for collections of discrete data such as a corpus. 6, 8, 23, 33, 36, 38, 39, 41, 52–54, 56, 57, 67, 68, 71, 80, 81, 84–86, 89, 90

Lsk

Lesk. Scientist who introduced an algorithm for word sense disambiguation based on words definitions overlapping. 30, 45, 54, 59

lso

Lowest super ordinate. The most specific common subsumer of two or more nodes. 30

NLP

Natural Language Processing. An hybrid field originated from the joint work of modern linguistics and AI researchers; it is also known as computational linguistics. 1, 3, 4, 6, 7, 12, 21

PDF

Probability Density Function. A function which can be integrated to obtain the probability that a random variable takes a value in a given interval. 38

PMF

Probability Mass Function. A function that defines the probabilities that a random variable takes particular values in its' range. 37, 38

PMI

Pointwise Mutual Information. A measure of association used in information theory and statistics. 23

RD

Reverse Dictionary. A dictionary where the user look up definitions and find words. 71, 79–81, 84, 86

tf

Term frequency. A measure that indicate how frequently a term occurs in a document. 10

tf·idf

Term frequency - inverse document frequency. A measure that indicates how important a term is achieved by weighing down the frequent terms while scaling up the rare ones. 10, 23

TID

The Integral Dictionary. A lexical database of French words. 11, 12

TSSI

Thesaurus Semantic Space I. The semantic space proposed for the distributional thesaurus with word vectors based on WordNet top concepts. 49, 51, 56, 62, 65–67

TSSII

Thesaurus Semantic Space II. The semantic space proposed for the distributional thesaurus with word vectors based on dynamic topics. 51, 56, 62, 65–67

VSM

Vector Space Model. An algebraic model for representing objects as vectors. 21–25