Coverage-Driven Refinement of Conceptual Representations

Haggai Toledano
Coverage-Driven Refinement of Conceptual Representations

Research Thesis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Haggai Toledano

Submitted to the Senate of the Technion — Israel Institute of Technology
Av 5772 Haifa July 2012
This research was carried out under the supervision of Prof. Shaul Markovitch, in the Faculty of Computer Science.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Shaul Markovitch, for his guidance and support throughout the years of my Masters degree studies. For ideas, discussions, constructive criticism and encouragement, without which this work would never have been realized.

Many thanks to Kira Radinsky and Sonya Liberman for many good hours of talks, of both personal and professional issues, over a cup of tea, and for being there when I needed them.

Thanks to my friends, Yochay Tzur, Danny Albocher, Shir Ben-Israel and Gal Naimark, for letting me win in billiards. To Noa Elgrably, Anat Hashavit and Sivan Bercovici, for lifting my spirit when I needed it. To my fellow Learning and Reasoning Laboratory members, Omer Levy, Erez Karpas, Ofer Egozi, Assaf Glazer and Ariel Raviv for their feedback and suggestions. And to all my other friends over the years of my studies for making them pass quicker and with more fun.

I am extremely grateful to Yonat Cohen, my girlfriend, who has always been there for me, understanding, supportive, cheering, knowing what words I needed to hear and when just to give a hug.

Endless thanks to my parents, Shlomo and Edith Toledano, for raising me up to believe in myself and for their unconditional love and support, and to my sister Noa, for her support for me and for her help with taking care of our cats.

And finally special thanks to my cats, whose purrs and snuggles were and still are an extremely effective method for relaxation.

The generous financial help of the Technion is gratefully acknowledged.
To Hita, who for the last three weeks of her short life sat and slept in my lap, bringing me comfort with her purrs, snuggles and joy of life, and giving me the strength to renew my dedication to this work just when I was on the verge of giving up on it.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>v</td>
</tr>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>Abbreviations and Notations</td>
<td>3</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Conceptual Representations</td>
<td>6</td>
</tr>
<tr>
<td>1.3 ESA and CHESA</td>
<td>8</td>
</tr>
<tr>
<td>2 Coverage and Noise in Conceptual Representations Stemming from Hierarchical Knowledge Repositories</td>
<td>11</td>
</tr>
<tr>
<td>2.1 The Limitations of ESA and CHESA</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Overcoming the Limitations of ESA and CHESA</td>
<td>16</td>
</tr>
<tr>
<td>2.3 Defining Coverage in Terms of Diversity</td>
<td>19</td>
</tr>
<tr>
<td>2.4 Using Relevant Concepts</td>
<td>22</td>
</tr>
<tr>
<td>2.5 Combining Diversity and Relevance</td>
<td>23</td>
</tr>
<tr>
<td>3 Coverage-Driven Refinement of Conceptual Representations</td>
<td>25</td>
</tr>
<tr>
<td>3.1 Algorithm Requirements</td>
<td>25</td>
</tr>
<tr>
<td>3.2 CDESA Algorithm</td>
<td>26</td>
</tr>
<tr>
<td>3.3 Association Scores - a relevance function for concepts and words</td>
<td>29</td>
</tr>
<tr>
<td>3.4 CAA - an implicit concepts coverage function</td>
<td>30</td>
</tr>
<tr>
<td>3.5 Examples</td>
<td>31</td>
</tr>
<tr>
<td>3.5.1 Examining single and multi-word representations</td>
<td>31</td>
</tr>
<tr>
<td>3.5.2 The contribution of concepts added for increased coverage to relatedness between words</td>
<td>36</td>
</tr>
<tr>
<td>3.5.3 Revisiting previous examples of ESA and CHESA limitations</td>
<td>40</td>
</tr>
</tbody>
</table>
4 Empirical Evaluation

4.1 Experimental Methodology

4.1.1 Experimental Setup

4.1.2 Knowledge Repository

4.1.3 Datasets

4.1.4 Correlation Measures

4.2 Datasets Experiments and Results

4.2.1 Determining the Best Parameters for Each Dataset

4.2.2 The Effect of $\delta$ on Performance

4.2.3 CDESA Robustness with Respect to $k$ and $\delta$

4.3 Datasets Results Correlation

4.3.1 Examining Datasets Results Correlation

4.3.2 Learning Parameters for CDESA

5 Related Work

6 Conclusions

References

Hebrew Abstract
## List of Figures

2.1 ESA similarity vs top concepts for **Israel** and **Hebrew** .......................... 13
2.2 ESA similarity vs top concepts for **Kasparov** and **Chess** ......................... 14
2.3 CHESA significance/association distribution for **mouse** ........................... 15
2.4 CHESA similarity rank instability, single words ........................................... 17
2.5 CHESA similarity rank instability, multiple words ........................................ 18
2.6 Full hierarchy of **SWISS ARMY KNIFE** within Wikipedia .......................... 20
2.7 Full hierarchy of **SWISS ARMY KNIFE** within Wikipedia; partial of **POCKET KNIFE**, **MACGYVER** .................................................... 21

3.1 $\Delta_{CAA}$ and association scores in the representation of **cat** ...................... 35
3.2 $\Delta_{CAA}$ and association scores in the representation of **the cat chased the mouse around the house** ................................................................. 39
3.3 CDESA and ESA graphs for **Kasparov** and **Chess** ................................. 44
3.4 CDESA and CHESA graphs for **shoe** vs **sock-boot-sandal** ......................... 45

4.1 WS353 correlation to human judgment vs top $- k$ and $\delta$ .......................... 57
4.2 MTurk correlation to human judgment vs top $- k$ and $\delta$ ........................... 58
4.3 LEE50 correlation to human judgment vs top $- k$ and $\delta$ ........................... 59
4.4 CDESA WS353 dependency of performance on $\delta$ ....................................... 60
4.5 CDESA MTurk dependency of performance on $\delta$ ....................................... 60
4.6 CDESA LEE50 dependency of performance on $\delta$ ....................................... 61
4.7 CDESA Robustness with respect to $(k, \delta)$ ............................................. 61
4.8 Correlation of results between WS353$_{Spearman}$ and MTurk$_{Spearman}$ ........ 62
4.9 Correlation of results between WS353$_{Spearman}$ and LEE50$_{Pearson}$ .......... 62
4.10 Correlation of results between MTurk$_{Spearman}$ and LEE50$_{Pearson}$ .......... 63
4.11 Spearman Correlations of Results .......................................................... 64
4.12 Visual correlation of Pearson and Spearman ............................................. 65
# List of Tables

2.1 Examples for ESA’s lack of abstraction ............................... 12

3.1 Sample concepts from the representation of *cat* .................. 33

3.2 Concepts from the representation of *cat* with $\Delta_{CAA} \geq 20$ .......... 34

3.3 Sample concepts from the representation of *the cat chased the mouse around the house* .................................. 37

3.4 Concepts from the representation of *the cat chased the mouse around the house* with $\Delta_{CAA} \geq 20$ .............................. 38

3.5 10 most highly associated common concepts of Ireland$_{1000}$ and Dublin$_{1000}$ 41

3.6 10 most highly associated common concepts in $(Ireland_{1000+}^{\geq 7} \cap Dublin_{1000+}^{\geq 7})$ that are not in $(Ireland_{1000} \cap Dublin_{1000})$ ............... 42

4.1 Wikipedia Statistics .................................................... 49

4.2 Best results for datasets (grid search) ............................... 52

4.3 Results of learning from one dataset ................................. 55

4.4 Results of learning from two datasets ............................... 56

4.5 Results of learning from datasets - summary ........................ 56
Abstract

Many text processing tasks are based on estimating semantic relatedness between texts. For example, in information retrieval, the relevancy of documents can be determined based on the semantic distance from the query. Recently, many algorithms have been developed for evaluating semantic relatedness based on a conceptual representation of the input texts. The concept spaces for these algorithms are based, in most cases, on large repositories of knowledge, such as Wikipedia and WordNet. Through these repositories, such representations are able to use more natural concepts and semantic relations than previous statistical corpora analysis based methods. The large concept spaces often yield representations that consist of very large collections of concepts. In many cases this has a negative impact on the performance of the semantic tasks due to redundancy that gives a superficially large weight to less relevant concepts, thus hiding important semantic aspects of the texts. In this work we present a new algorithm that produces semantic interpretations of texts in the form of conceptual representations which are based on hierarchical concept spaces. The algorithm incrementally adds strongly-associated concepts to the representation, while using the hierarchical structure of the semantic database to maximize coverage. Inherent to this algorithm is the problem of finding an acceptable trade-off between concept coverage, enabling a more detailed semantic interpretation of the texts, and concept redundancy which degrades the performance of semantic tasks. We suggest a solution to this problem, that uses the hierarchical structure of the semantic database to compute a stopping condition to the algorithm. We test the new algorithm for text relatedness tasks and show its advantage over existing approaches.
## Abbreviations and Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDESA</td>
<td>Coverage-Driven Explicit Semantic Analysis</td>
</tr>
<tr>
<td>ESA</td>
<td>Explicit Semantic Analysis</td>
</tr>
<tr>
<td>CHESA</td>
<td>Compact Hierarchical Explicit Semantic Analysis</td>
</tr>
<tr>
<td>TF.IDF</td>
<td>Term Frequency x Inverse Document Frequency</td>
</tr>
<tr>
<td>CAA</td>
<td>Concept and Ancestors</td>
</tr>
<tr>
<td>$CAA(v, G)$</td>
<td>$v \bigcup \cup_{v' \in \text{parents}(v)} CAA(v', G)$</td>
</tr>
<tr>
<td>$CAA(S, G)$</td>
<td>$\cup_{v \in S} CAA(v, G)$</td>
</tr>
<tr>
<td>$\Delta_{CAA}(v, S, G)$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$v$ candidate node to add, $S$ current representation, $G$ concept graph</td>
</tr>
<tr>
<td>$\Delta_{CAA}$</td>
<td>$\Delta_{CAA}(v, S, G)$ in context</td>
</tr>
<tr>
<td>$k$</td>
<td>CDESA parameter to include $k$ most highly associated concepts</td>
</tr>
<tr>
<td>$\delta$</td>
<td>CDESA parameter that defines the minimal $\Delta_{CAA}$ that constitutes a new subject</td>
</tr>
<tr>
<td>text</td>
<td>A string</td>
</tr>
<tr>
<td>CONCEPT</td>
<td>A concept from the knowledge repository</td>
</tr>
<tr>
<td>* CONCEPT</td>
<td>A concept that is a category in the knowledge repository</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Today's age of information needs tools for the automatic processing of information. A large portion of the world's information is encoded as text (Lyman & Varian, 2003), which is easier to produce and to process than images, recorded sounds or other forms of media. Texts account for encyclopedic knowledge, news reports, blogs, personal information and many other incarnations of information. A vast amount of textual information is generated each moment around the world (Hilbert, 2012), and tools to find relevant texts are essential. General-purpose search engines such as Google, Bing and Yahoo! are used daily by millions of people to search the Internet (comScore, 2010), medical literature has its own search engine in PubMed, and practically any website, be it a store, a forum or an other type of website, incorporates a textual search engine to help its users find the information they seek.

Besides encoding texts in some inner structure for easier access, information retrieval tools require a measure of relevancy of these texts to user queries. While many such queries can be answered by simple string matching (Aho & Corasick, 1975) (e.g. the query *apples* will yield all the texts that contain the pattern *apples* exactly or its stemmed form *apple*), this may not always suffice (Lenat, Guha, Pittman, Pratt, & Shepherd, 1990). Sometimes relevancy is meant as semantic relatedness, in which case a query must also yield results that do not contain the query pattern itself but texts that are related by meaning alone (e.g. the query *apple* will also yield texts about pears (both are kinds of fruits) or computers (Apple is a computers company) even if the pattern *apple* is not found in them). We use the term *related* rather than *similar* following Budanitsky and Hirst (2006) and Gabrilovich (2006), who argued that semantic relatedness is a more appropriate term for various natural language processing tasks since they seek not only similarity relations such as synonyms and hyponyms between texts, but also relations that similarity does not cover, such as meronymy, antonymy and entailment among other possible relations.

Methods that represent texts as a collection of data that arises only from patterns
that exist within the texts themselves (e.g. pattern matching (Ravichandran & Hovy, 2002), word counts (e.g. Bag of Words) (Baeza-Yates, Ribeiro-Neto, et al., 1999)) generally have quite a good performance. However, they have a disadvantage compared with methods that use extra-textual information relevant to the texts in question, quite simply because they do not have this information (Hjørland, 1998). This is especially true for short texts or for words, where the objects to compare do not have enough information in them for a computer algorithm to realize the relations between them. This extra-textual information can relate texts semantically in ways much like humans relate them, i.e. by abstraction, finding synonyms or antonyms (e.g. by using WordNet (Budanitsky & Hirst, 2006)), disambiguation of words sense (Yarowsky, 1995) or syntactic structures (Jurafsky, 1996) by training on a corpus, etc.; or by representing them using a corpus in a manner that while carrying semantic meaning it does so in a way that is undecipherable to humans, e.g. Latent Semantic Analysis (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990).

One of the prominent approaches for evaluating semantic relatedness is by using large ontologies of general (i.e. non domain-specific) knowledge. By mapping texts onto these ontologies, reasoning about the semantic relatedness of the mapped texts can be made through reasoning on the similarity of their corresponding mappings. A popular source for such ontological knowledge is Wikipedia (http://www.wikipedia.org/), an online encyclopedia which articles have mostly textual data (e.g. in (Gabrilovich & Markovitch, 2007), (Liberman & Markovitch, 2009), (Zesch, Gurevych, & Mühlhäuser, 2007)). Other similar ontologies include the Open Directory Project, human edited hierarchical directory of the Internet (http://www.dmoz.org/) in e.g. (Maguitman, Menczer, Roinestad, & Vespignani, 2005); WordNet, a lexical database of English, where words are interlinked by conceptual-semantic and lexical relations (http://wordnet.princeton.edu/) in e.g. (Patwardhan & Pedersen, 2006); Cyc, a database of human 'common sense' (http://www.cyc.com/ and others.

While the discussion above concentrates on representations of texts for information retrieval, semantic representations of texts are also used in word sense disambiguation (Navigli, 2009), machine translation (Beale, Nirenburg, & Mahesh, 1995), human-machine interfaces (Cuayáhuital, Dethlefs, Richter, Tenbrink, & Bateman, 2010), question answering (Dali, Rusu, Fortuna, Mladenic, & Grobelnik, 2009), document categorization (Wang, Hu, Zeng, & Chen, 2009) and a host of other tasks that involve the processing of human created texts according to their meaning (rather than just their structure such as lossless file compressing).

1.2 Conceptual Representations

One approach of semantically representing texts is by conceptual representations - a representation of texts in which the semantics of the texts are defined (possibly not exclusively) by semantic entities called concepts.
A concept in the context of this work is an atomic pointer to a real world object (physical or abstract) or to an event. This object or event carries a certain semantic information. For example consider the object apple (fruit). It carries the semantic information of what the fruit apple is, which can be what an apple fruit is physically (color, dimensions, chemical composition, etc.), methods for growing apples, the importance of apples in human culture and similar semantic data. The collection of the semantic information the object carries is referred to as its semantic meaning or meaning.

However, from the viewpoint of the algorithm that uses the concept that corresponds with apple (fruit), the meaning of apple (fruit) is not important. It only uses the concept itself, not its corresponding object or object’s meaning, for the conceptual representation. Therefore a concept’s name is of no actual importance to the algorithm that uses it, as concepts are just mappings from placeholders to objects or events; e.g. the concept apple (computer company), although similar in form to the concept apple (fruit) (both contain the word apple), is semantically different; they could equally be named obj#45 and obj#97. However, for the sake of convenience and clarity, we will refer to concepts by the names of the objects or events they point to.

In this work a conceptual representation of a text (single or multi word) is defined as a set of concepts and a measure of their relevancy to the text. For example, the text apples grow in temperate and subtropical climates may be represented by the (partial) list of concepts and their (qualitative) relevancy scores (higher scores mean more relevant concepts) - ( (apple (fruit), 7.3), (fruit, 4.5), (climate, 2.3), (apple (company), 0.2), (cellular phone, 0.001) ). If the representation is complete (i.e. it contains all the concepts available in the knowledge repository) it is often very sparse as most concepts are not relevant to most texts - texts, short or long, usually discuss only a very small set of subjects. While many concepts may be relevant to each subject, this is still a relatively small portion of the concepts provided by the knowledge repository that being large and comprehensive covers a multitude of unrelated subjects, with mostly disjoint corresponding concept subsets.

The concepts can be generated from knowledge repositories that hold a term-to-knowledge relationship. For example, with an encyclopedic knowledge repository, its articles can be considered as concepts and their textual content as the knowledge the concepts carry; with a lexical dictionary as knowledge repository, words can be the concepts and their definitions and semantic functions are the knowledge the concepts carry. It is important to note that the concepts are not related to any specific dataset on which the representation algorithm operates, but represent an independent world knowledge that may be far more comprehensive than the represented texts themselves.
1.3 ESA and CHESA

ESA (Explicit Semantic Analysis) (Gabrilovich & Markovitch, 2007) and CHESA (Compact Hierarchical Explicit Semantic Analysis) (Liberman & Markovitch, 2009) are two conceptual representation methods that use Wikipedia (primarily; but they can also use other data sources) as their extra-textual knowledge base.

ESA represents texts by mapping them to concepts, giving each concept a score that defines its relevance to the represented text. Using Wikipedia as an example for the knowledge source (and similarly for other data sources), it uses articles as concepts and the articles’ textual content to calculate the concepts’ relevance to texts. At the preprocessing stage ESA assigns each Wikipedia article a vector of the words it contains, with each word’s score set to TF.IDF with respect to its occurrence in an article relative to its occurrence in other Wikipedia articles. Each article is considered a concept, and its vector is used to construct representations of words in the following manner: ESA constructs a vector of the TF.IDF scores the represented word received for each article it appears in. Thus each word is represented as a vector of association scores to concepts, henceforth concept vector. Multi-word texts are represented as the vector sum of the representations of the words they are composed of. The full representation can then be trimmed to its top X features (those with the highest association score), where X is set manually; however as this has proven to not be beneficial to text relatedness tasks, the full representation is usually used.

CHESA aims to provide a small but concise representation, as opposed to the extremely large one generated by ESA. In order to achieve that it uses the hierarchical nature of the knowledge source to filter out concepts that are of lower relevance to the represented text. Its representations are similar to ESA’s, but they add concepts that correspond to categories in the knowledge base hierarchy. Again, using Wikipedia as an example for the knowledge source, CHESA traverses the Wikipedia hierarchy in such a way that it always works on a front of a partial hierarchy of Wikipedia, rooted at Wikipedia’s top category (of which all other categories and articles are descendants). It iteratively adds nodes to the representation using a measure of local significance to determine which node to add next - the node with the highest significance score is selected (there is also a bottom-up variant of the algorithm which removes nodes from the full representation in an ascending order of significance score). This significance measure attempts to assess the impact a node has on the representation by measuring the additional information it gives relative to that of its parent, with respect to the represented text. This process continues until the desired hierarchy size is achieved.

For calculating association scores, a method similar to ESA’s TF.IDF is used for words within an article or category with respect to their global prevalence in Wikipedia (the content of an article is its text, the content of a category is the union of the contents of its descendants). Thus the representation is a hierarchy of concepts derived from both the leaves and the inner nodes of the hierarchy, with each having an association score to...
the represented text. For multi-word texts CHESA offers two alternatives: one similar to ESA - the representation of a multi-word text is the sum of the representations of the individual words of the text; the other method utilizes the single word CHESA method for the full text by using, instead of how many times a word appears in a concept’s content, how many times a subset of the text’s words appear in that content.

Both ESA and CHESA use cosine similarity\(^1\) between concept vectors to assess the relatedness of their respective texts. For ESA this is straightforward as its output is a concept vector. For CHESA, the hierarchy is first flattened to a concept vector by using each node in the hierarchical representation as an element of the concept vector. Then the cosine similarity can be calculated.

While both ESA and CHESA utilize Wikipedia to represent texts and achieve high correlation to human judges on several text relatedness tasks, they do have some short-comings which prevent them from performing better: They both produce noisy representations; ESA representations lack abstractions while CHESA representations are too abstract; and CHESA representations show a large instability of relatedness ranks with respect to representation size. These shortcomings are discussed in detail in section 2.1 of this work.

ESA was a breakthrough algorithm in its use of Wikipedia and other such ontologies for semantic relatedness. Being the state-of-the-art algorithm for the semantic interpretation of texts, it was used as the basis of many natural language processing algorithms, and is still considered the golden standard for the performance of semantic relatedness tasks. Therefore addressing its limitations may benefit its many descendant algorithms.

We address the limitations of ESA and CHESA with a conceptual representation algorithm called Coverage-Driven Explicit Semantic Analysis (CDESA) that aims to balance between noise and coverage of concepts by utilizing hierarchical knowledge repositories to create representations that are composed of a relatively small set of relevant concepts that cover a diverse array of subjects. It does so by using both association scores and structural properties of the hierarchical repository to select relevant and diverse concepts.

The rest of this work is organized as follows: In chapter 2 we analyze in detail the limitations of ESA and CHESA and provide an outline of our solution to these limitations. Chapter 3 presents the CDESA algorithm and shows how it improves upon the limitations of ESA and CHESA. In Chapter 4 we evaluate the performance of CDESA on several experimental setups. Chapter 5 covers related works. We conclude the work in chapter 6.

\[^1\] \( \text{sim}(\vec{v}_1, \vec{v}_2) = (\vec{v}_1 \cdot \vec{v}_2) / (||\vec{v}_1|| \cdot ||\vec{v}_2||) \)
Chapter 2

Coverage and Noise in Conceptual Representations Stemming from Hierarchical Knowledge Repositories

In the scope of this work, the coverage of a conceptual representation is a measure of the amount of concepts that are included in the representation relative to the amount of all the concepts available from the knowledge source. The higher the coverage of a representation, the more information it holds, so it would intuitively seem that algorithms that use higher coverage representations would perform better.

However, this intuition ignores the observation that too much information may actually degrade performance. This is due to a factor called noise, referring to many concepts that are not very related to the represented text and are very weak in individual effect in the representation, but combined they bias the representation in such a way that it does not truly represent its associated text anymore.

In this chapter we analyze the limitations of ESA and CHESA in the context of coverage and noise, and present the basis of our solution to these problems.

2.1 The Limitations of ESA and CHESA

We begin by an analysis of the limitations of ESA and CHESA.

- **ESA does not abstract well**: ESA does not take into account the hierarchical structure of Wikipedia. It uses only articles as features, and does not use categories. This leads to a loss of abstraction, as categories are abstractions of articles and categories of higher levels are abstractions of categories themselves (a cat is a member of DOMESTICATED ANIMALS, which is a member of ANIMALS, in itself a member of ZOOLOGY, etc.).
Abstraction is essential for the proper determination of the relatedness of texts. Although ESA uses abstraction implicitly by considering all the articles, in particular all the articles that belong to any certain category, it may fail to capture the semantic relatedness of texts if they do not evoke the same concepts but their relationship is only shown in higher level categories. Thus the lack of abstraction hides similarities between the semantics of texts.

For example, consider the words **flashlight** and **aluminium**. Although they are highly related (many flashlights are made from aluminum), they share very few evoked articles (i.e. articles that contain both the word **flashlight** and the word **aluminium**). Their similarity score is 0.0078, based on 10 articles common to 494 articles for **flashlight** and 1969 articles for **aluminium**. However, when categories are added, many more concepts are common, and the terms’ association score is more than quadrupled to 0.0329, based on 1124 common concepts (from a total of 3379 for **flashlight** and 7263 for **aluminium**). A higher similarity score of the concept vectors means a higher relatedness of the terms they represent. Common categories include **metallurgy**, **hiking equipment** and **electrical wiring**. See table 2.1 for additional examples.

### Table 2.1: Examples for ESA’s lack of abstraction - common concepts with articles only vs common concepts with articles plus categories, for pairs of words of different relations

<table>
<thead>
<tr>
<th>Word 1</th>
<th>Word 2</th>
<th>Concepts Count</th>
<th>Similarity Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Articles Only</td>
<td>Articles + Categories</td>
</tr>
<tr>
<td><strong>flashlight</strong></td>
<td><strong>aluminium</strong></td>
<td>10 (494, 1969)</td>
<td>1124 (3379, 7263)</td>
</tr>
<tr>
<td><strong>fork</strong></td>
<td><strong>lunch</strong></td>
<td>11 (1179, 3597)</td>
<td>2148 (5345, 15310)</td>
</tr>
<tr>
<td><strong>computer</strong></td>
<td><strong>keyboard</strong></td>
<td>165 (5391, 2063)</td>
<td>1746 (12732, 5777)</td>
</tr>
<tr>
<td><strong>man</strong></td>
<td><strong>woman</strong></td>
<td>382 (5172, 3720)</td>
<td>6287 (16964, 14032)</td>
</tr>
<tr>
<td><strong>cat</strong></td>
<td><strong>animal</strong></td>
<td>509 (4193, 5700)</td>
<td>5948 (15453, 14913)</td>
</tr>
<tr>
<td><strong>apple</strong></td>
<td><strong>knife</strong></td>
<td>30 (2705, 4232)</td>
<td>4280 (10715, 16435)</td>
</tr>
<tr>
<td><strong>beach</strong></td>
<td><strong>sunset</strong></td>
<td>90 (3381, 1340)</td>
<td>2899 (11108, 6662)</td>
</tr>
</tbody>
</table>

- **ESA representations are noisy**: ESA representations contain association scores for all Wikipedia’s concepts. While this may offer more information for the represented texts, when comparing texts using cosine similarity or other additive similarity measures the many less associated concepts may mask the few more associated concepts, thus biasing the relatedness measures the wrong way. Selecting the most highly associated concepts as a way to overcome this difficulty, e.g. in (Egozi, Gabrilovich, & Markovitch, 2008), could be problematic for two reasons: Firstly, the most highly associated concepts are usually overly specific, and have a lower probability to be selected for any two representations, yielding lower and less robust similarity scores. Secondly, while the most highly associated concepts
Figure 2.1: Cosine similarity vs Top most associated concepts in ESA for Israel and Hebrew: X axis is the size limit of a representation (Israel tops at 4985 non-zero concepts, Hebrew tops at 3184). At most, the two words’ representations share 1429 non-zero concepts. The Y axis shows concept co-occurrence in both representations and the similarity score of the representations as the representation size limit grows.

carry the main senses of a text, selecting only them might prevent including secondary senses, which in turn might prevent detecting relatedness between texts that are related only with regard to their secondary meanings. This is shown in figure 2.1: selecting only the the $X$ most highly associated concepts may lead to low cosine similarity between the representations, because for small values of $X$ (smaller than a few hundreds) there are very few common concepts for both representations.

On the other hand, selecting too many concepts for the representation may introduce concepts related to less relevant secondary senses of the texts, thus unjustifiably lowering the similarity score even if the main senses of the words are very highly related. Figure 2.2 shows that unlike the example in figure 2.1, selecting too many highly associated concepts may actually reduce the relatedness score of representations, because the represented texts are similar in their primary sense but dissimilar in their secondary sense.

- CHESA’s representations are too abstract: In CHESA, less associated concepts with high local significance are included because of the representation building method: A concept is added to the representation if it contributes to the representation significantly more than its parent with respect to the text be-
Figure 2.2: Cosine similarity vs Top most associated concepts in ESA for Kasparov and Chess: X axis is the size limit of a representation (Kasparov tops at 194 non-zero concepts, Chess tops at 2914). At most, the two words’ representations share 174 non-zero concepts. The Y axis shows concept co-occurrence in both representations and the similarity score of the representations as the representation size limit grows. Note the lower scores after the 1000 size limit mark. Common concepts that contributed to the lower score include mainly biographical data about Kasparov (e.g. BAKU, AZERBAIJANI JEWS) but also chess related articles which are not particularly related to Kasparov (e.g. MSN GAMES).

For the following example, a CHESA representation of the word mouse was calculated with four sizes. The graphs in figure 2.3 show the association and significance scores of each concept in the representations. The progress of representation size is reflected in the growth of the number of concepts with low significance scores (recall that concepts are added to the representation in a descending or-
Figure 2.3: Significance/Association scores distribution for representations of mouse

- **CHESA’s representations are too unstable:** Representation size in CHESA is a parameter to the algorithm, and is a very important aspect of CHESA, as the algorithm’s purpose was to provide a compact representation (as opposed to...
As in ESA, the representation size affects the similarity score of representations. However, CHESA’s instability with respect to representation size is larger. This is because representation size in ESA limits the representation to the top-X concepts with respect to their association scores, so enlarging the representation size adds concepts with lower scores, affecting the cosine similarity score to a lesser degree. With CHESA, the representation size limits the hierarchy most abstract concepts number, as discussed in the previous bullet, so additions of concepts with high association scores that affect the cosine similarity score to a higher degree are more prevalent.

While this is expected, it has a problematic side effect - different representation sizes induce different rank order of similarity scores between pairs of words. Consider the following example: What word is the more related to the word sock from the following two words - shoe or sandal? Figure 2.4(a) shows the similarity scores of the pairs sock-shoe and sock-sandal with respect to representation size. The two pairs are ranked most related (highest similarity score) about the same number of times for the representation sizes tested. This means there is no clear indication for which pair is more related, an unwanted property of the algorithm. This instability phenomenon is harmful both for correlation scores that are based on rank, such as used for several datasets, and those that rely on absolute score; see chapter 4 (specifically section 4.1.4) for more details.

More examples for this phenomenon can be seen in figures 2.4(b) and 2.5.

To summarize: ESA and CHESA have several drawbacks - not enough abstraction in ESA; too much abstraction in CHESA; instability of the representation with respect to size limits - all of these are related to unsatisfactory dealing with noisy representations. While there is certainly a need for the inclusion of abstract concepts, the topmost nodes in the hierarchy may not provide the best results. Also, while there is a need for the inclusion of relevant concepts, those with the highest association scores may not suffice. This calls for a middle way: Inclusion of category concepts for better inclusion of abstract senses of a text, and selection of informative concepts that include both strong and weak senses of the text in a way that is less obstructive to additive similarity measures such as cosine similarity (which were shown to perform better than non-additive measures such as the Jaccard measure in (Liberman, 2010)).

2.2 Overcoming the Limitations of ESA and CHESA

In this work we deal with conceptual representations that stem from hierarchical knowledge repositories (e.g. Wikipedia). This allows an easy extraction of concepts - leaves and inner nodes (in Wikipedia these are articles and categories) and of abstraction relationships - a node is an abstraction of its descendants (in Wikipedia - a category is an abstraction of its sub-categories and the articles that it and they include). A concep-
Figure 2.4: CHESA single word semantic relatedness vs representation size: Is sock more related to shoe or to sandal? And what is the relatedness order among the pairs shoe-sock, shoe-boot and shoe-sandal?
(a) Word pairs: Cosine similarity between gold rings and silver rings with respect to representation size. Note the large changes in cosine similarity even for reasonably large representation sizes.

(b) Long texts: Cosine similarity between the opening sentences of the Wikipedia articles sand cat (The sand cat (Felis margarita), also referred to as the 'sand dune cat', is a small wild cat distributed over African and Asian deserts) and jungle cat (The jungle cat (Felis chaus) is a medium-sized cat native to Asia from southern China in the east through Southeast and Central Asia to the Nile Valley in the west) with respect to representation size. The instability in similarity score is less pronounced than with shorter texts (for representations sizes over 200), however it is still present.

Figure 2.5: CHESA multi-word similarity vs representation size

tual representation would then be a set of the hierarchy’s nodes and association scores between them and the represented text. From here on, the term node will be used to describe the hierarchical properties of a concept, and the term concept will be used to describe content related properties of a concept.

As discussed in the previous section, conceptual representations need to balance between coverage and noise - maximal coverage with minimal noise. For coverage maximization, we aim to include as many concepts as possible from those our knowledge repository provides. For noise minimization we aim to not include concepts that provide non-relevant information. For this subset selection task for a given representation and a target function (e.g. for a representation \( R \) with concepts set \( C \), calculate \( \text{select}(R) = \text{argmax}_{V \subseteq C} \text{coverage}(RV) - \text{noise}(RV) \)), in essence an exhaustive search of the optimal set of concepts should be made. However, given the huge amount of concepts from any knowledge repository that covers a reasonable amount of human knowledge, this task is unfeasible. We therefore use a heuristic approach to address the problem.

### 2.3 Defining Coverage in Terms of Diversity

In its most simplistic form, the coverage of a subset of concepts from a representation may be defined by the percentage of concepts the subset includes with respect to the complete set of those in the representation. This, however, completely disregards different types of subsets - those that cover a coherent semantic set (all concepts belong to a single subject), those that cover a diverse semantic set (multiple subjects), those that give an emphasis to a certain semantic set over the others. We address the previously discussed problem of coverage vs noise balancing by defining a coverage measure that is geared towards subject diversity. This way, maximizing concept coverage results in a less redundant representation than with the simplistic definition of coverage. This is done using the fact that as the conceptual representation stems from a hierarchical knowledge repository, the hierarchical structure can be used to determine which type the subset belongs to. This in itself is not enough, and excess coverage is addressed in sections 2.4 and 2.5.

A hierarchical knowledge repository is constructed by humans and so its structure represents the grasp humans have of the relations between its nodes. Its leaves contain data regarding a specific subject (in an encyclopedic repository the leaves are articles containing text and sometimes images, sounds, videos, etc.). When several leaves are grouped under a parent node, they represent a family of data that deals with aspects of a certain subject, represented by the parent node. This parent node is then an abstraction of these leaves. Inner nodes in the hierarchy can then be grouped together under a higher level parent node, which is an abstraction of its children and their descendants. The children of a node may be a mixture of inner nodes and leaves. Subjects can often be related to several fields (e.g. a calculator is an office tool but also an
American invention), so each node may have multiple parents. An example of the tree of concepts from Wikipedia’s root to Swiss Army Knife is presented in figure 2.6. Figure 2.7 adds partial hierarchies for the Pocket Knife and MacGyver articles to illustrate the inner nodes and leaves mixture (Pocket Knife is an article which is a sister node to the category Camping Equipment; the MacGyver partial hierarchy is used to illustrate leaves at different depths - it is in depth 4 from the root while both Swiss Army Knife and Pocket Knife are in depth 6).

Figure 2.6: Full hierarchy of Swiss Army Knife within Wikipedia
Created using GraphViz through http://www.pinboard.jp/mp/webdot
Figure 2.7: Full hierarchy of SWISS ARMY KNIFE within Wikipedia; partial hierarchies of POCKET KNIFE and MACGYVER
Created using GraphViz through http://www.pinboard.jp/mp/webdot
A concept (a node) belongs to a certain subject that is its path from the hierarchy root. In the Swiss Army Knife example, the concept Simple Living belongs both to Society→Social Movements and to Belief→Spirituality. We say that if a concept belongs to a certain subject, then it implicitly covers its subject, hence a representation that contains the concept Simple Living implicitly covers Society, Social Movements, Belief and Spirituality, and also Main Topic Classifications. This implicit subject coverage is important: it reflects how much abstract information a certain concept conveys. If a representation includes a certain concept that implicitly covers a certain subject, adding another concept that implicitly covers the same subject to the representation might add redundant information. Consider, for example, a representation of the word knife that contains the concept Swiss Army Knife (because the article Swiss Army Knife contains the word knife). Adding the concept Simple Living (its descendant articles contain the word knife) will not add another subject to the representation (although Simple Living may be related differently to knife from how Swiss Army Knife is related to knife, as indicated by different association scores to those concepts in the representation). Also, adding the concept Pocket Knife, although not a parent of Swiss Army Knife, will not add much information to the representation for a similar reason. If, on the other hand, a representation contained only the concept Simple Living then concepts like Tools and Weapons would add to the implicit subject coverage of the representation by contributing a coverage of subjects not covered before.

While trying to assess whether a certain set of concepts provides a good enough coverage of the full representation, we should remember our target: to provide a representation that is as much noise-free as possible. Using multiple concepts that implicitly cover the same subject may contribute noise to the representation, as they are just repeating already existing information, although with slight differences.

Using the above mentioned reasoning, we can use the measure of implicit subject coverage to measure the subject diversity of a set of concepts, preferring representations that are more diverse in their subjects over representations that are more uniform in their subjects.

### 2.4 Using Relevant Concepts

While subject diversity is important, it is equally important to include relevant concepts in the representation.

A concept’s relevancy to the represented text is judged by a measure called association score which is a measure to how much the text is found in the concept relative to how much it is found in the entire knowledge repository (very similar to TF.IDF). The exact details can be found in chapter 3. The higher the association score, the more relevant to the text the concept is.

If we include concepts to account only for diversity, we might be again subjected
to noise. This time because both relevant and non-relevant concepts will receive equal
importance in the representation. Also, because generally there are much more less-
relevant concepts in a representation, the less relevant concepts will have even more
power and bias the representation to the less important senses of the text. Having a
large weight to secondary senses could be beneficial if we compare only texts that are
related only in their secondary senses, but this is a rare case. In most cases we need
to rank related texts according to the concepts we deem most accurately describe the
main senses of the texts, with lower consideration to the secondary senses.

Since the main senses of a text are represented by the concepts with the highest
association score, it is beneficial to include those in the representation. However, usually
the first few highest associated concepts are too specific, so in order to achieve a better
coverage of relevant concepts, and a better chance of representations having concept
co-occurrence of their main senses, additional highly associated concepts should be
included.

Consider the following example: For the word "knives", the concepts with highest
association scores are the concepts KNIVES OUT and NIGHT OF KNIVES with the concept
KNIFE appearing only third highest; these are followed by several very specific knife
related concepts like knives for specific uses (e.g. SURVIVAL KNIFE at 11th highest
association score, UTILITY KNIFE at 21), knife manufacturers (e.g. SPYDERCO at 19,
VICTORINOX at 33); knife related items are scarce (e.g. CAN OPENER at 11, FORK
at 365) and knife making materials do not appear before BULAT STEEL at 326. These
concepts may not contribute much in terms of diversity, however they could be very
helpful to differentiate similar knife related concepts.

2.5 Combining Diversity and Relevance

We have reasoned that representations that are diverse and contain relevant concepts
are our target. To achieve this we have yet to solve one last issue, which is the problem
of selecting the concepts that fulfill the diversity and relevance requirements. This is
again a problem of subset selection, which with hundreds of thousands of concepts is
an infeasible task to perform accurately.

Selecting the most relevant concepts is an easier task, just select the $X$ concepts
with the highest association score. $X$ can be determined based on a sample dataset or
simply as a parameter to the algorithm.

Selecting concepts so as to diversify the representation is a more complex task.
Recall the definition of how diversity is measured - a concept contributes to diversity
if its addition to the representation increases the size of the implicit subject coverage
of the representation. The problem is that this amount is dependent on the concepts
already found in the representation.

Consider figure 2.6. Suppose we would like to select only concepts that implicitly
cover more than 3 concepts (i.e. we decided that a concept that implicitly covers three
previously unseen concepts constitutes a new subject), and that concepts are considered to be added to the representation one at a time:

- Scenario I: The representation consists only of the concept MECHANICAL HAND TOOLS. Then it implicitly covers HAND TOOLS, TOOLS and all the predecessors of TOOLS, for a total of 7 concepts (including MAIN TOPIC CLASSIFICATIONS). Testing for diversity contribution of the concept KNIVES, we see that KNIVES implicitly covers CUTTING TOOLS and its predecessors, including the previously covered TOOLS and its predecessors, for a total of 13 concepts. In this scenario, KNIVES adds 7 concepts to the implicit coverage, therefore we consider it contributing to add it to the representation.

- Scenario II: The representation consists only of the concept KNIVES. Then it implicitly covers CUTTING TOOLS and all of its predecessors for a total of 13 concepts. Testing for diversity contribution of the concept MECHANICAL HAND TOOLS we find that it implicitly covers 7 concepts, but only one of them (HAND TOOLS) was not implicitly covered by the representation, so we consider the addition of MECHANICAL HAND TOOLS to be non-contributing to the representation.

At the end of scenario I the representation contains both the concept MECHANICAL HAND TOOLS and the concept KNIVES, but at the end of scenario II the representation contains only the concept KNIVES. All that has changed between the two scenarios is the order in which the concepts are considered for addition to the representation.

A possible solution is to ignore order, and to add a concept based only on its implicit coverage, disregarding other already added concepts. But in this case, all concepts whose depth is over some predetermined value will be added (in the above example case, if their depth is bigger than 3), and the representation will be noisy.

The solution selected for this work is: Consider the addition of concepts to the representation in a descending order of association score. This heuristic is used so that high coverage concepts that are also more relevant to the text being represented are preferred over concepts with similar coverage that are less relevant.

Exactly how the algorithm works is covered in chapter 3.
Chapter 3

Coverage-Driven Refinement of Conceptual Representations

In this section we present a new algorithm for the semantic interpretation of texts. The algorithm takes as input a text consisting of one or more words and a hierarchical set of concepts, and produces a concise representation that covers the various semantic aspects of the text on one hand while avoiding redundancy on the other hand.

The algorithm operates on a single text (i.e. a text’s representation is independent of other texts’ representations), and is unsupervised with global parameters. This has the following properties:

- **Representation independence**: Each text is represented independently. This means a shorter processing time for representing each text (no need to consider data other than the text itself) and greater scalability (linear in the number of texts). This is important for applications that demand processing of a massive amount of texts, such as many information retrieval tasks do.

- **Unsupervised operation**: While the algorithm is parametrized and can be adapted to different text types as needed, it does not need any tagged data for its operation. This is an advantage in environments where such data is unavailable or hard to obtain (e.g. a new data set in which relations between texts are unknown). It saves the time consuming and imprecise operation of tagging by humans, which has the additional drawback that it needs to be calculated on a representative sample of the data set - a hard task when the data set size is measured in millions or billions of items.

3.1 Algorithm Requirements

The algorithm proposed in this work has the following requirements:

- **An input text**: A natural language text $T$, composed of one or more words.
• **A hierarchy of concepts:** A directed acyclic graph (DAG) $G(V,E)$ with a single root, in which each leaf is associated with a text in the language of the input text. Internal nodes should not be associated with a text of their own, but are assumed to be associated with the set of texts of their descendant nodes.

• **An association score function between concepts (hierarchy nodes) and texts:** $\tilde{v} = \text{assoc}(G, v \in V, T)$. This will be used to calculate the numerical values of concepts for the final representation vector $C$.

• **An implicit concepts coverage function:** A function $\langle v, G \rangle \rightarrow \text{cover}(v, G)$ that returns the implicit concept coverage of $v$ within $G$ (i.e. the ancestors of $v$ in $G$), as explained in chapter 2.

To fulfill the above mentioned requirements, one can simply use the vector $C(T) = \bigcup_{v \in V} \langle v, \text{assoc}(G, v, T) \rangle$ as a representation of $T$. This is the full concept vector which, as explained in previous chapters, may lead to worse results than using a subset $F \subseteq V$ and calculating the representation vector as $C(T) = \bigcup_{v \in F} \langle v, \text{assoc}(G, v, T) \rangle$.

### 3.2 CDESA Algorithm

The algorithm we propose is called **Coverage-Driven Explicit Semantic Analysis**, and will be henceforth abbreviated to CDESA. It calculates a full conceptual representation of a text, and then refines it so that its concepts provide a good coverage of the text’s senses while not being too noisy so as to provide a diverse representation with many relevant concepts. To achieve this, the algorithm is parametrized with two parameters: $k$ - the number of most relevant concepts to include in the representation, and $\delta$ - the number of new concepts that if covered by a certain concept constitute a new subject (see section 2.5 for an example of the use of $\delta$). The algorithm is further parametrized so that the user, in addition to selecting $k$ and $\delta$, can decide on the knowledge hierarchy to use, the association score calculation method and the coverage calculation method.

The word algorithm (algorithm 1) represents single-word texts in context, while the text algorithm (algorithm 2) represents multi-word texts and uses algorithm 1 as its basis. Both algorithms return a concept vector of the word or text they were given. Note that for the representation of single words with no context in algorithm 1, $T = w$. Thus, in practice, one can use algorithm 2 for both words and for texts and regard algorithm 1 as a sub-procedure.

The word algorithm works as follows: For a given word $w$ to represent within the context of a text $T$, the representation is initialized with the $k$ concepts with the highest association scores (within the context of $T$) to the word $w$. The algorithm then scans all the remaining available concepts from the concept hierarchy in a descending order of association scores (within the context of $T$) to $w$. Whenever an implicit coverage of a scanned concept includes at least $\delta$ concepts that are not in the combined implicit
coverage of the current representation, the concept is added to the representation. The algorithm returns the set of \( \langle \text{concept}, \text{association score} \rangle \) for concepts added to the representation. The text algorithm, for a given text \( T \) to represent, combines the representations of \( w \in T \) calculated using the word algorithm, and returns a set of \( \langle \text{concept}, \text{association score} \rangle \) for concepts from these representations.

In algorithm 2, a text’s representation is defined as a combination of the representations of the words it is composed of. It does so with disregard to sentence structure (including word order), similarly to ESA and CHESA. However, the way it combines the representation differs from ESA and CHESA:

- Both ESA and CHESA use vector summation for representing texts: The association score of a concept \( v \) in the representation of \( T \) is the sum of \( v \)'s association scores in the representations of the words \( w \) in \( T \). This gives a large weight to concepts that have a large association score in many word representations. While this is reasonable, as these are the concepts that represent the words (and thus the text) best, their combination does not need to be linear. An \( L_2 \) norm might be a better choice - highly relevant individual word concepts would still receive a high relevance score in the text representation, but their strength relative to that of the other concepts would be lower. This is needed in order to allow concepts that receive high scores in a small number of word representations to still be pronounced in the text representation, as they may represent an important aspect of it, even if it was not repeated in many words of the text.

- CHESA offers another method for multi-word text representation: Treating the entire text as a single unit, calculating the relevance of a concept according to all the words from the text it contains rather than according to each word separately, considering only concepts that include at least \( \ln(|T|) \) words from \( T \). If \( T \) is a word, then \( |T| = 1 \) and \( \ln(|T|) = 0 \), so all concepts that contain the word \( T \) are included in the representation. If, for example, \( |T| = 50 \), then only concepts that contain at least 4 words from \( T \) \( (\ln(|T|) = 3.9) \) will be included in the representation of \( T \). This way only concepts that are in the general context of \( T \) are included in its representation. This method focuses the representation on concepts that are more related to the text, as they contain several words from it rather than just one - it keeps the representation in the context of the text. This method is good in focusing the representation on relevant concepts, but has a problem: It disregards multiple occurrences of words in the text, so if some word is more prevalent in \( T \), although it might be the main topic of \( T \) and we would want a higher score for its concepts, this method does not provide this distinction.

To use the best of all worlds, CDESA scores concepts for each word separately and combines them using the \( L_2 \) norm. It discards concepts if they contain less than \( \ln(|T|) \) words from \( T \).
This way, CDESA produces representations that are text-context aware, while bal-
ancing between concepts that are highly related to many words from the text and
concepts that are highly related to only few words from the text.

In addition, CDESA returns only concepts which include words from the represented
text. This is just an optimization on memory, since the full representation’s size is in
the order of hundreds of thousands of concepts, while most of them are totally unrelated
to the text - they (or their descendants) do not contain words from it. Removing these
concepts does not change cosine similarity, as with an association score of 0 they neither
affect the dot product nor the vector size calculations. For the sake of the completeness
of the representation, concepts that do contain words from the represented text but are
given a 0 association score (elevated to 0 because of under-representation, see section
3.3) are kept in the representation, since they do bear some relation to the represented
text.

Algorithm 1

Algorithm CDESA\((w,T,G,assoc,k,\text{cover},\delta)\) - Single Word

Require:

- \(w\) - the word to represent
- \(T\) - a context text
- \(G(V,E)\) - a directed acyclic graph
- \(\langle v \in V, w \rangle \rightarrow assoc(G,v,w)\) - a relevance function from concepts to words
- \(k \leq |V|\) - a minimal number of most relevant concepts to include
- \(\langle S,G \rangle \rightarrow \text{cover}(S,G)\) - an implicit concept coverage function
- \(\delta \leq |V|\) - a minimal number of concepts that constitute a subject for implicit
  subject coverage

1: \(V' \leftarrow k\) concepts from \(V\) with highest \(assoc\) value that contain at least \(\lceil \ln(|T|) \rceil\)
words from \(T\)
2: \(P \leftarrow V'\)
3: repeat
4: \(v \leftarrow \) the concept with highest \(assoc\) value in \(V \setminus P\) that contains at least \(\lceil \ln(|T|) \rceil\)
words from \(T\)
5: if \(\text{cover}(V' \cup \{v\}, G) - \text{cover}(V', G) \geq \delta\) then
6: \(V' \leftarrow V' \cup \{v\}\)
7: end if
8: \(P \leftarrow P \cup \{v\}\)
9: until \(P = V\)
10: return \(\cup_{v \in V'}\langle v, assoc(G,v,T)\rangle\)
Algorithm 2 Algorithm CDESA(T,G,assoc,cover,δ) - Multiple Words

Require:

• T - a multi-word text to represent
• G, assoc, k, cover, δ - parameters as in algorithm 1

1: for all \( w_i \in T \) do
2: \( \tilde{w}_i \leftarrow \text{CDESA}(w_i, T, G, \text{assoc}, k, \text{cover}, \delta) \)
3: end for
4: \( P \leftarrow \bigcup \{ v_j | \langle v_j, r_j \rangle \in \tilde{w}_i \} \)
5: for all \( v \in P \) do
6: \( \tilde{v} \leftarrow \sqrt{\sum_j \{ r_i^2 | \langle v, r_i \rangle \in \tilde{w}_j \}} \)
7: end for
8: return \( \bigcup_{v \in P} \langle v, \tilde{v} \rangle \)

3.3 Association Scores - a relevance function for concepts and words

The relevance score a concept is given with respect to the represented word is called an
association score. It describes how relevant the concept is to the word, with respect
to how relevant the entire hierarchy is to the word.

To calculate the association score of the word \( w \) and a concept \( v \), we use the formula
from (Liberman & Markovitch, 2009). First a relative frequency of \( w \) in \( v \) is computed:

\[
f(w, v) = \frac{\text{count}(w, v)}{\sum_{w_j \in v} \text{count}(w_j, v)}
\]

\( \text{count}(w, v) \) counts the occurrences of \( w \) within \( v \) and all its descendants in the concept hierarchy. Note that the sparseness of concept vectors stems from this calculation, as for most concepts \( v \), \( f(w, v) \) would be 0 since most words do not occur in most concepts. The function \( f(w, v) \) is called the term frequency of \( w \) in \( v \).

The association score \( \tilde{v}_w \) of \( v \) and \( w \) is then:

\[
\tilde{v}_w = \max \left( 0, \log \left( \frac{f(w, v)}{f(w, r)} \right) \right)
\]

where \( r \) is the root concept of the hierarchy. A negative \( \log \left( \frac{f(w, v)}{f(w, r)} \right) \) means \( \frac{f(w, v)}{f(w, r)} < 1 \), indicating an under-representation of \( w \) in \( v \) relative to how it is represented in the entire graph.

For the CDESA algorithm, we define the function \( \text{assoc}(G, v, w) = \tilde{v}_w \).
3.4 CAA - an implicit concepts coverage function

For defining the implicit concepts coverage function, used to determine implicit subject coverage, we define the following function: Given a graph $G(V,E)$ and a node $v$,

$$CAA(v,G) = v \cup \bigcup_{v' \in \text{parents}(v)} CAA(v',G)$$

$CAA(v,G)$ is the set of $v$ and its ancestors in $G$. For a set of nodes $S$:

$$CAA(S,G) = \bigcup_{v \in S} CAA(v,G)$$

or the set of nodes in $S$ and their ancestors in $G$. $|CAA(S,G)|$ can be used as the cover function for CDESA.

The CDESA algorithm uses the cover function to determine whether a concept contributes a new subject to the representation at hand. To investigate how this works, we define a function that mimics the way CDESA uses the cover function:

$$\Delta_{CAA}(v,S,G) = |CAA(S \cup \{v\},G)| - |CAA(S,G)|$$

$\Delta_{CAA}(v,S,G)$ is the additional implicit concept coverage of $v$ in $G$ if added to $S$. A large value of $\Delta_{CAA}$ means that $v$ would add a large subject to $S$. A small value of $\Delta_{CAA}$ means that $v$ would add a small sized subject (i.e. a subject with very few concepts) to $S$ (either $S$ already covers a large part of that subject, or the subject is small by nature). $\Delta_{CAA} = 0$ means $v$ (and its ancestors) are already included in the implicit cover of $S$.

A low value of $\Delta_{CAA}$ needs to be discussed further. It can occur in one of two situations:

- $S$ already covers a large part of the subject implicitly covered by $v$. This is the normal situation and here $\Delta_{CAA}$ expresses what we want - $v$ would not contribute much to implicit coverage if added to $S$.

- The subject covered by $v$ is small by nature, i.e. it would contribute very few implicit concepts even to an empty representation, or in other words, the reverse tree of $v$ within $G$ is small. This is a tricky spot - if the subject is small but not yet covered, it is not immediate to dismiss it as irrelevant to the represented text, maybe it is relevant but the implicit coverage will never include it because it is small. This may happen with e.g. low depth nodes, which generally have very few ancestors (top categories in Wikipedia, for instance, only have one ancestor, Wikipedia’s root). However, in CDESA the $\Delta_{CAA}$ measure is used to add concepts only if their association score is low but they can still contribute a previously unseen subject to the representation. If a subject is small and its representative concept has a low association score, then it can be considered as
noise and not be added to the representation. So $\Delta_{CAA}$ actually conforms with the noise reduction theme of CDESA, therefore it is reasonable to use it in it as it is.

It is important to note that CDESA using the $\Delta_{CAA}$ method will almost always add leaves to the representation, rather than inner nodes. This is because it scans nodes in a descending order of association scores, and every inner node has at least one content node (a leaf) which has a higher or equal association score than its own. Thus on most cases (practically all cases which affect similarity scores) the leaves are scanned before their ancestors. If a leaf is added then its ancestors are marked as having been implicitly covered, and would not be added to the representation by themselves. If the leaf was not added, then it did not pass the $\Delta_{CAA} \geq \delta$ test, and so all of its ancestors would not pass it too (since their $CAA$ is a subgraph of the leaf’s $CAA$). The result is that besides extreme cases in which the scan order is not defined (all the association scores of nodes in an inner node’s subtree are equal; this mostly happen when they are all zeros), CDESA adds only leaves using the $\Delta_{CAA}$ method. Note that adding an inner node does not prevent its descendants to be added later on, if they satisfy the algorithm’s conditions for addition to the representation; however if an inner node is added before its descendants, they will have a lower $\Delta_{CAA}$ score than if it was not added.

3.5 Examples

To illustrate how the algorithm works in real life, we present a few examples, all based on Wikipedia as a knowledge repository.¹

3.5.1 Examining single and multi-word representations

We begin by examining nodes with various values of $\Delta_{CAA}$, using the representation of the word cat as an example.

Table 3.1 shows the ten concepts with the highest association scores from the representation of the word cat. The table shows for each concept its name, $\Delta_{CAA}$ value, association score, meaning and the concepts affecting its $\Delta_{CAA}$ value. As $\Delta_{CAA}$ is calculated for concepts in a descending order of association scores, concepts affecting a certain concept’s $\Delta_{CAA}$ score can only be concepts with higher association scores than its own. The effect of previous concepts on a certain concept’s $\Delta_{CAA}$ value is their shared ancestors, the set of which is deducted from a concept’s ancestors set to calculate its $\Delta_{CAA}$. The shared ancestors value is written in parenthesis after each affecting concept.

¹The version used is the same as used in (Liberman & Markovitch, 2009), which is a snapshot of Wikipedia taken on October 18th, 2007

31
The concept **cat (unix)** is the most highly associated concept, and therefore its \( \Delta_{CAA} \) score is not affected by other concepts and is its full set of ancestors union the concept itself. It adds the subject of “Unix commands” to the representation. The second concept by order of association scores is **cat**. It shares only one ancestor with **cat (unix)** (the category *nature*), since it adds a completely new subject to the representation - “A type of feline”. Likewise, **cheshire cat** adds the subject of “fictional cats” and **plasan sand cat** adds the subject of “military vehicles”. **Cool cat**, a cartoon, shares many concepts related to fiction and entertainment with **cheshire cat**; it also adds the subject of “cartoons”, including its film aspects. **Claude cat** is a cartoon character and its entire set of ancestors is covered by the ancestors of **cheshire cat** and **claude cat**, therefore **claude cat** does not contribute any new subjects to the representation other than the concept itself. Similar behavior is witnessed with the other concepts in the table.

As the representation algorithm progresses, and tests concepts with lower association scores, more and more previous concepts are available to affect the \( \Delta_{CAA} \) of each concept, and so concepts with lower association scores have a lower probability of having high \( \Delta_{CAA} \) values. Table 3.2 demonstrates this phenomena by displaying all the concepts from the representation of **cat** with \( \Delta_{CAA} \geq 20 \). The association score rank of each concept indicates its position in the representation’s concepts, ordered by descending association scores. Note the relative rarity of concepts as the association rank increases. There are still a few concepts that contribute new subjects to the representation, such as **john cunningham (raf officer)**, related to **cat** through his nickname, “cat’s eye”, and adding the subjects of “World War II”, “British Decorations”; and **robert a. heinlein**, related to **cat** through his novel “The Cat Who Walks Through Walls”, and adding the subject of “Libertarian Politics” to the representation. The size of the representation of **cat** is 15453, and the last concept with \( \Delta_{CAA} \geq 20 \) is **robert a. heinlein** at rank 4982, meaning that the two thirds of the set of concepts in the representation all have lower than 20 \( \Delta_{CAA} \) value.

Figure 3.1 shows the progress of the representation algorithm by displaying the \( \Delta_{CAA} \) values and association scores for all the concepts in the representation of **cat**. Subfigure 3.1(a) shows the raw values of \( \Delta_{CAA} \) and subfigure 3.1(b) shows their exponential moving average for better visualization of their trend. Note the general decline of \( \Delta_{CAA} \) values with lower association scores and the spikes in \( \Delta_{CAA} \) values indicating new subjects discovered, growing rarer and lower as the algorithm progresses.

An example of how a multi-word text is represented: For this example we examine the representation of the phrase **the cat chased the mouse around the house**. It exhibits a very similar behavior to that of the above discussed representation of **cat**. Table 3.3 shows the ten most highly associated concepts of the representation of the phrase with their meaning and concepts they share ancestors with; table 3.4 shows concepts from the representation whose \( \Delta_{CAA} \) value is equal or larger than 20; figure 3.2 shows the representation algorithm’s progress though the values \( \Delta_{CAA} \) and the
<table>
<thead>
<tr>
<th>Association Rank</th>
<th>Concept Name</th>
<th>( \Delta_{CAA} )</th>
<th>Association Score</th>
<th>Concept Meaning</th>
<th>Concepts Affecting ( \Delta_{CAA} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CAT (UNIX)</td>
<td>11</td>
<td>8.38</td>
<td>A Unix command</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>CAT</td>
<td>18</td>
<td>8.00</td>
<td>A type of feline</td>
<td>CAT (UNIX) (1)</td>
</tr>
<tr>
<td>3</td>
<td>CHESHIRE CAT</td>
<td>18</td>
<td>7.92</td>
<td>A fictional cat</td>
<td>CAT (UNIX) (1) CAT (2)</td>
</tr>
<tr>
<td>4</td>
<td>PLASAN SAND CAT</td>
<td>15</td>
<td>7.88</td>
<td>A military vehicle</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>COOL CAT</td>
<td>11</td>
<td>7.88</td>
<td>A cartoon</td>
<td>CAT (2) CHESHIRE CAT (8)</td>
</tr>
<tr>
<td>6</td>
<td>CLAUDE CAT</td>
<td>1</td>
<td>7.82</td>
<td>A cartoon character</td>
<td>CAT (2) CHESHIRE CAT (9) COOL CAT (17)</td>
</tr>
<tr>
<td>7</td>
<td>BIG CAT</td>
<td>6</td>
<td>7.82</td>
<td>A term: large feline</td>
<td>CAT (1)</td>
</tr>
<tr>
<td>8</td>
<td>CAT’S EYE (FILM)</td>
<td>9</td>
<td>7.76</td>
<td>A film</td>
<td>COOL CAT (7) CLAUDE CAT (7)</td>
</tr>
<tr>
<td>9</td>
<td>CATS &amp; DOGS</td>
<td>4</td>
<td>7.75</td>
<td>A film</td>
<td>CAT (1) CHESHIRE CAT (1) COOL CAT (8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLAUDE CAT (8) CAT’S EYE (FILM) (10)</td>
</tr>
<tr>
<td>10</td>
<td>.CAT</td>
<td>29</td>
<td>7.74</td>
<td>A file extension</td>
<td>CAT (UNIX) (1) CAT (1) CHESHIRE CAT (2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>COOL CAT (1) CLAUDE CAT (1) CATS &amp; DOGS (1)</td>
</tr>
<tr>
<td>Association Score Rank</td>
<td>Concept Name</td>
<td>$\Delta_{CAA}$</td>
<td>Association Score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------</td>
<td>----------------</td>
<td>-------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>.cat</td>
<td>29</td>
<td>7.74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Leopard Cat</td>
<td>43</td>
<td>7.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Cat Grant</td>
<td>22</td>
<td>7.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Top Cat</td>
<td>52</td>
<td>7.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Jacob Cats</td>
<td>24</td>
<td>7.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Schrödinger’s cat</td>
<td>24</td>
<td>7.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Black cat</td>
<td>23</td>
<td>7.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Cat Deeley</td>
<td>29</td>
<td>7.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Fat Cat (Rescue Rangers)</td>
<td>21</td>
<td>7.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>Black Cat, White Cat</td>
<td>32</td>
<td>7.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>Cat Power</td>
<td>21</td>
<td>7.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>Cat Whitehill</td>
<td>20</td>
<td>6.94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>Cat Stevens</td>
<td>39</td>
<td>6.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>Black Cat (comics)</td>
<td>25</td>
<td>6.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>Cat Cora</td>
<td>20</td>
<td>6.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>280</td>
<td>Uncaria tomentosa</td>
<td>24</td>
<td>6.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>393</td>
<td>John Cunningham (RAF officer)</td>
<td>31</td>
<td>5.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1321</td>
<td>Dr. Seuss</td>
<td>22</td>
<td>4.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1805</td>
<td>Emir Kusturica</td>
<td>22</td>
<td>4.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2090</td>
<td>Ernie Ladd</td>
<td>21</td>
<td>3.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3758</td>
<td>Fredric Brown</td>
<td>21</td>
<td>3.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3977</td>
<td>Mick McCarthy</td>
<td>21</td>
<td>2.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4049</td>
<td>Michael Chertoff</td>
<td>28</td>
<td>2.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4056</td>
<td>Bill Frist</td>
<td>20</td>
<td>2.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4091</td>
<td>A. L. Rowse</td>
<td>24</td>
<td>2.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4953</td>
<td>Margaret Atwood</td>
<td>33</td>
<td>2.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4982</td>
<td>Robert A. Heinlein</td>
<td>20</td>
<td>2.40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Concepts from the representation of `cat` with $\Delta_{CAA} \geq 20$
Categories are marked with *
Figure 3.1: $\Delta_{\text{CAA}}$ and association scores in the representation of cat
association scores of the concepts.

Note that thanks to the context-sensitive nature of the representation algorithm, namely that represented words from a phrase need to be in context with each other, the resulting representation is much more focused than that of cat’s, having only cartoon-related concepts in its top-10 list. Also note that the category * mickey mouse television series has $\Delta_{CAA} = 0$, because it is the ancestor of a previous concept, disney’s house of mouse, thus it cannot contribute any new subject to the representation, not even itself - it is already completely implicitly covered by a previous concept. Curiously, the concept mickey’s magical christmas: snowed in at the house of mouse does not share any ancestors with previous concepts - the previous concepts discussing “Mickey Mouse” cover only the television cartoons aspect of the subject, while the concept mickey’s magical christmas covers only its film aspects.

Note also that the $\Delta_{CAA}$ and association scores diminish much faster for the multi-word text than for the single-word one above, and also that the representation size is much smaller. This happens because of the context limiting requirement, namely that a concept would be included in the representation only if it contains more than $\ln(|T|)$ words from $T$; in this case (disregarding stop words) $\ln(|\{cat, chased, mouse, house\}|) = 1.4$, so each concept must contain at least two words from $\{cat, chased, mouse, house\}$ in order to be included in the representation. The general shapes of the graphs, however, are similar, indicating that the algorithm behaves similarly for both single words and multiple words texts.

3.5.2 The contribution of concepts added for increased coverage to relatedness between words

In this example we examine how concepts that were added because they had a high $\Delta_{CAA}$ value contribute to relatedness measures between representations.

Due to the need to limit representation size in order to reduce the noise of superfluous and lesser related concepts, as discussed earlier in chapter 2, the CDESA algorithm takes only the $k$ most highly associated concepts for a representation, then adds concepts whose $\Delta_{CAA}$ is equal to or larger than $\delta$ in order to compensate for the coverage lost resulting from taking only the top-$k$ concepts. In this section we show what is the benefit of adding concepts with $\Delta_{CAA} \geq \delta$

Consider the relatedness between Ireland and Dublin. Dublin is the capital of Ireland, and they are related in terms of politics, history, climate, etc. For full representations, Ireland and Dublin share 5784 concepts. The top-1000 representations (i.e. 1000 most highly associated concepts of the full representations) however share only 132 common concepts. We define the top-1000 representations of Ireland and Dublin as Ireland$_{1000}$ and Dublin$_{1000}$ respectively.

The most commonly highly associated concepts (measured as the multiplication of the association scores of common concepts) of Ireland$_{1000}$ and Dublin$_{1000}$ are not
<table>
<thead>
<tr>
<th>Association Score Rank</th>
<th>Concept Name</th>
<th>$\Delta_{CAA}$</th>
<th>Association Score</th>
<th>Concept Meaning</th>
<th>Concepts Affecting $\Delta_{CAA}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A MOUSE IN THE HOUSE</td>
<td>14</td>
<td>15.90</td>
<td>Tom &amp; Jerry cartoon</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>PET PEEVE (TOM AND JERRY)</td>
<td>15</td>
<td>14.58</td>
<td>Tom &amp; Jerry cartoon</td>
<td>A MOUSE IN THE HOUSE (12)</td>
</tr>
<tr>
<td>3</td>
<td>JOHANN MOUSE</td>
<td>9</td>
<td>13.74</td>
<td>Tom &amp; Jerry cartoon</td>
<td>A MOUSE IN THE HOUSE (12) PET PEEVE (TOM AND JERRY) (15)</td>
</tr>
<tr>
<td>4</td>
<td>DISNEY’S HOUSE OF MOUSE</td>
<td>22</td>
<td>13.64</td>
<td>Mickey Mouse cartoon</td>
<td>PET PEEVE (TOM AND JERRY) (8) JOHANN MOUSE (4)</td>
</tr>
<tr>
<td>5</td>
<td>OLD ROCKIN’ CHAIR TOM</td>
<td>1</td>
<td>13.55</td>
<td>Tom &amp; Jerry cartoon</td>
<td>A MOUSE IN THE HOUSE (12) PET PEEVE (TOM AND JERRY) (12) JOHANN MOUSE (12)</td>
</tr>
<tr>
<td>6</td>
<td>PUSS GETS THE BOOT</td>
<td>1</td>
<td>13.37</td>
<td>Tom &amp; Jerry cartoon</td>
<td>A MOUSE IN THE HOUSE (12) PET PEEVE (TOM AND JERRY) (12) JOHANN MOUSE (14) DISNEY’S HOUSE OF MOUSE (2) OLD ROCKIN’ CHAIR TOM (12)</td>
</tr>
<tr>
<td>7</td>
<td>* MICKEY MOUSE TELEVISION SERIES</td>
<td>0</td>
<td>13.37</td>
<td>Mickey Mouse cartoon</td>
<td>PET PEEVE (TOM AND JERRY) (2) JOHANN MOUSE (3) DISNEY’S HOUSE OF MOUSE (10) PUSS GETS THE BOOT (2)</td>
</tr>
<tr>
<td>8</td>
<td>TRAP HAPPY</td>
<td>1</td>
<td>13.29</td>
<td>Tom &amp; Jerry cartoon</td>
<td>A MOUSE IN THE HOUSE (12) PET PEEVE (TOM AND JERRY) (12) JOHANN MOUSE (12) OLD ROCKIN’ CHAIR TOM (12) PUSS GETS THE BOOT (12)</td>
</tr>
<tr>
<td>9</td>
<td>LITTLE SCHOOL MOUSE</td>
<td>1</td>
<td>13.09</td>
<td>Tom &amp; Jerry cartoon</td>
<td>A MOUSE IN THE HOUSE (12) PET PEEVE (TOM AND JERRY) (12) JOHANN MOUSE (12) OLD ROCKIN’ CHAIR TOM (12) PUSS GETS THE BOOT (12) TRAP HAPPY (12)</td>
</tr>
<tr>
<td>10</td>
<td>MICKEY’S MAGICAL CHRISTMAS: SNOWED IN AT THE HOUSE OF THE MOUSE</td>
<td>12</td>
<td>13.02</td>
<td>Mickey Mouse cartoon</td>
<td>—</td>
</tr>
<tr>
<td>Association Score</td>
<td>Rank</td>
<td>Concept Name</td>
<td>∆CAA</td>
<td>Association Score</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
<td>------------------------------------------</td>
<td>------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>Disney’s House of Mouse</td>
<td>22</td>
<td>13.64</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>43</td>
<td>Mighty Mouse</td>
<td>43</td>
<td>12.75</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>20</td>
<td>House Mouse</td>
<td>20</td>
<td>12.15</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>26</td>
<td>Mouse Trap</td>
<td>26</td>
<td>11.38</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>21</td>
<td>John Congleton</td>
<td>21</td>
<td>11.23</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>20</td>
<td>Big Cat Week</td>
<td>20</td>
<td>10.77</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>23</td>
<td>Fat Cat (Rescue Rangers)</td>
<td>23</td>
<td>10.37</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>31</td>
<td>Mickey Mouse</td>
<td>31</td>
<td>10.20</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>33</td>
<td>Margay</td>
<td>33</td>
<td>10.16</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>22</td>
<td>Neverball</td>
<td>22</td>
<td>10.10</td>
<td></td>
</tr>
<tr>
<td>233</td>
<td>42</td>
<td>Dr. Seuss</td>
<td>42</td>
<td>9.35</td>
<td></td>
</tr>
<tr>
<td>235</td>
<td>39</td>
<td>Rumer Godden</td>
<td>39</td>
<td>9.32</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>22</td>
<td>Dune II</td>
<td>22</td>
<td>9.30</td>
<td></td>
</tr>
<tr>
<td>281</td>
<td>24</td>
<td>The Third Man</td>
<td>24</td>
<td>8.90</td>
<td></td>
</tr>
<tr>
<td>294</td>
<td>25</td>
<td>Cheetah</td>
<td>25</td>
<td>8.81</td>
<td></td>
</tr>
<tr>
<td>331</td>
<td>27</td>
<td>Donoghue v Stevenson</td>
<td>27</td>
<td>8.64</td>
<td></td>
</tr>
<tr>
<td>464</td>
<td>26</td>
<td>Wario</td>
<td>26</td>
<td>7.96</td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>26</td>
<td>Minerva McGonagall</td>
<td>26</td>
<td>7.82</td>
<td></td>
</tr>
<tr>
<td>508</td>
<td>26</td>
<td>Brian Setzer</td>
<td>26</td>
<td>7.81</td>
<td></td>
</tr>
<tr>
<td>532</td>
<td>21</td>
<td>Giorgio Moroder</td>
<td>21</td>
<td>7.67</td>
<td></td>
</tr>
<tr>
<td>544</td>
<td>23</td>
<td>Red Squirrel</td>
<td>23</td>
<td>7.61</td>
<td></td>
</tr>
<tr>
<td>573</td>
<td>20</td>
<td>Chris Ware</td>
<td>20</td>
<td>7.50</td>
<td></td>
</tr>
<tr>
<td>633</td>
<td>20</td>
<td>Robert Stephen Hawker</td>
<td>20</td>
<td>7.21</td>
<td></td>
</tr>
<tr>
<td>663</td>
<td>35</td>
<td>Lin Biao</td>
<td>35</td>
<td>7.07</td>
<td></td>
</tr>
<tr>
<td>675</td>
<td>23</td>
<td>Günter Grass</td>
<td>23</td>
<td>7.02</td>
<td></td>
</tr>
<tr>
<td>723</td>
<td>28</td>
<td>Walt Disney</td>
<td>28</td>
<td>6.75</td>
<td></td>
</tr>
<tr>
<td>733</td>
<td>37</td>
<td>Fredric Brown</td>
<td>37</td>
<td>6.69</td>
<td></td>
</tr>
<tr>
<td>1015</td>
<td>20</td>
<td>Tim Curry</td>
<td>20</td>
<td>5.46</td>
<td></td>
</tr>
<tr>
<td>1039</td>
<td>29</td>
<td>Tracey Ullman</td>
<td>29</td>
<td>5.35</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4: Concepts from the representation of the cat chased the mouse around the house with ∆CAA ≥ 20
Categories are marked with *
Figure 3.2: $\Delta_{CAA}$ and association scores in the representation of the cat chased the mouse around the house
necessarily those that are highly associated in any of the individual representations. This is shown in table 3.5: for each concept its association rank in each representation is displayed. For example, the concept Dublin is ranked first in the representation of Dublin but only 565th in the representation of Ireland. Note that while association scores are high, some concepts have high $\Delta_{CAA}$ values while others have low such values.

Now we consider a minimum additional implicit concept coverage of $\Delta_{CAA} \geq 7$ to constitute a new subject in CDESA (parameter $\delta = 7$). Let $\text{Ireland}_{1000+\geq7}$ be concepts from the representation of Ireland that are either in $\text{Ireland}_{1000}$ or have a $\Delta_{CAA}$ value of 7 or more. Similarly define $\text{Dublin}_{1000+\geq7}$. Then table 3.6 shows the 10 most highly associated common concepts in ($\text{Ireland}_{1000+\geq7} \cap \text{Dublin}_{1000+\geq7}$) that are not in ($\text{Ireland}_{1000} \cap \text{Dublin}_{1000}$). This table shows the benefit of including concepts with high $\Delta_{CAA}$ value - some concepts exist only in the top 1000 associated concepts of one representation, so they are relevant to it. However they are not in the top 1000 list of the other representation so they would not be included in any relatedness measure between the representations unless added with the $\Delta_{CAA}$ method.

What does $\Delta_{CAA} \geq 7$ means in this example? Take for example the concept OPERATION LOBSTER I (a Nazi infiltration attempt to Ireland in 1940). It appears in the representation of Dublin in rank 1017, so it did not get into the list of Dublin$_{1000}$. It did however get in the list of Ireland$_{1000}$ in rank 969. Its implicit coverage in the representation of Dublin are the following categories: CANCELED MILITARY OPERATIONS INVOLVING GERMANY, MILITARY OPERATIONS INVOLVING GERMANY, MILITARY HISTORY OF GERMANY, HISTORY OF GERMANY, WORLD WAR II WESTERN EUROPEAN THEATRE, WORLD WAR II EUROPEAN THEATRE, CAMPAIGNS AND THEATRES OF WORLD WAR II, MILITARY OPERATIONS OF WORLD WAR II. This means the representation of Dublin did not include any concept that implicitly covered the above categories, so this is an aspect of Dublin that would not have been accounted for if the $\Delta_{CAA}$ method would not have been used.

This example shows how the coverage in terms of diversity allows CDESA to account for relevant concepts while reducing the noise by not including all concepts beyond a certain relevancy point (in the example - beyond the 1000th most highly associated concept) but still including representatives of important subjects, that could help in identifying relationships to other representations otherwise undetectable.

3.5.3 Revisiting previous examples of ESA and CHESA limitations

In section 1.3 we presented several examples of the shortcomings of ESA and CHESA. We can now test how CDESA handles them. Here we show that CDESA’s representations are more stable than ESA’s and CHESA’s - they handle the noise of large representations better than ESA and they are better similarity order-preserving than CHESA over various representation sizes.
<table>
<thead>
<tr>
<th>#</th>
<th>Concept Name</th>
<th>Rank in Ireland</th>
<th>Rank in Dublin</th>
<th>$\Delta_{CAA}$ in Ireland</th>
<th>$\Delta_{CAA}$ in Dublin</th>
<th>Association Score to Ireland</th>
<th>Association Score to Dublin</th>
<th>Association Scores Multiplied</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DUBLIN</td>
<td>565</td>
<td>1</td>
<td>1</td>
<td>37</td>
<td>4.58</td>
<td>9.04</td>
<td>41.42</td>
</tr>
<tr>
<td>2</td>
<td>PRIMATE OF IRELAND</td>
<td>388</td>
<td>18</td>
<td>2</td>
<td>26</td>
<td>4.80</td>
<td>7.55</td>
<td>36.20</td>
</tr>
<tr>
<td>3</td>
<td>* ANGLICAN ARCHBISHOPS OF DUBLIN</td>
<td>381</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>4.80</td>
<td>7.46</td>
<td>35.85</td>
</tr>
<tr>
<td>4</td>
<td>IRISH BEER</td>
<td>20</td>
<td>353</td>
<td>6</td>
<td>1</td>
<td>6.59</td>
<td>5.43</td>
<td>35.77</td>
</tr>
<tr>
<td>5</td>
<td>THE AUTOMOBILE ASSOCIATION (IRELAND)</td>
<td>97</td>
<td>217</td>
<td>2</td>
<td>2</td>
<td>5.93</td>
<td>5.79</td>
<td>34.33</td>
</tr>
<tr>
<td>6</td>
<td>TRANSPORT IN IRELAND</td>
<td>71</td>
<td>272</td>
<td>4</td>
<td>1</td>
<td>6.12</td>
<td>5.59</td>
<td>34.22</td>
</tr>
<tr>
<td>7</td>
<td>* CHURCH OF IRELAND</td>
<td>171</td>
<td>133</td>
<td>0</td>
<td>0</td>
<td>5.48</td>
<td>6.18</td>
<td>33.87</td>
</tr>
<tr>
<td>8</td>
<td>PARLIAMENTARY CONSTITUENCIES IN THE REPUBLIC OF IRELAND</td>
<td>54</td>
<td>381</td>
<td>1</td>
<td>1</td>
<td>6.27</td>
<td>5.36</td>
<td>33.61</td>
</tr>
<tr>
<td>9</td>
<td>RUGBY LEAGUE IRELAND</td>
<td>29</td>
<td>562</td>
<td>2</td>
<td>2</td>
<td>6.52</td>
<td>5.07</td>
<td>33.02</td>
</tr>
<tr>
<td>10</td>
<td>IRELAND WOMEN’S CRICKET TEAM</td>
<td>49</td>
<td>522</td>
<td>6</td>
<td>7</td>
<td>6.31</td>
<td>5.11</td>
<td>32.29</td>
</tr>
<tr>
<td>#</td>
<td>Concept Name</td>
<td>Rank in Ireland</td>
<td>Rank in Dublin</td>
<td>Association Score</td>
<td>Association Scores</td>
<td>Multiplied Score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-------------------------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SCOUTING IRELAND</td>
<td>67</td>
<td>1002</td>
<td>7</td>
<td>12</td>
<td>28.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>HIGH LANE</td>
<td>1005</td>
<td>197</td>
<td>11</td>
<td>71.3</td>
<td>24.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>EDMUND ALLEN MEREDITH</td>
<td>1017</td>
<td>328</td>
<td>10</td>
<td>3.96</td>
<td>5.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>THE TAKING OF CHRIST (Caravaggio)</td>
<td>1070</td>
<td>382</td>
<td>7</td>
<td>3.85</td>
<td>5.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>KEVIN IZOTH O'DOHERTY</td>
<td>570</td>
<td>1048</td>
<td>8</td>
<td>4.82</td>
<td>24.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>OLIVER PLUNKETT</td>
<td>1019</td>
<td>737</td>
<td>11</td>
<td>4.15</td>
<td>20.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>JOHN LAVERY</td>
<td>1059</td>
<td>584</td>
<td>8</td>
<td>5.05</td>
<td>19.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>THOMAS BUTLER (FOOTBALLER)</td>
<td>835</td>
<td>1029</td>
<td>5</td>
<td>4.38</td>
<td>19.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CATHAL GANNON</td>
<td>1133</td>
<td>319</td>
<td>7</td>
<td>3.50</td>
<td>19.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>OPERATION LOBSTER I</td>
<td>969</td>
<td>1017</td>
<td>1</td>
<td>4.29</td>
<td>19.18</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.6: 10 most highly associated common concepts in \( (\text{Ireland}_{1000} \cap \text{Dublin}_{1000} \geq 7) \) that are not in \( (\text{Ireland}_{1000} \cap \text{Dublin}_{1000}) \)
The first example we revisit is cosine similarity vs representation size for ESA. In figure 3.3 are shown graphs for CDESA (3.3(a), with $\delta$ fixed at 3) and for ESA (3.3(b), a copy of figure 2.2 for reference) for the pair Kasparov-Chess. The graph shows cosine similarity and common concepts vs representation size (which in CDESA signifies the $k$ most highly associated concepts). Although the CDESA representation includes many more concepts than ESA’s for each representation size, it reaches its peak cosine similarity at about the same representation size, and as opposed to ESA it remains stable at this cosine similarity value.

The second example we show is the relatedness rank preservation vs representation size for CHESA for single word pairs. In figure 3.4 are shown the graphs for CDESA (3.4(a), $\delta = 3$) and for CHESA (3.4(b), a copy of figure 2.4(b) for reference) for the three pairs shoe-sock, shoe-boot and shoe-sandal. The graph shows cosine similarity vs representation size (which in CDESA signifies the $k$ most highly associated concepts). Here we show that CDESA is much more consistent than CHESA in its cosine similarity rank, and that for representations sizes larger than 200, compared to 5 alterations of rank order in the CHESA representations CDESA alters the ranks only once.

These examples show that there are cases where CDESA is more consistent than ESA and CHESA. In other cases, such as the examples of Israel-Hebrew (figure 2.1), and shoe vs sock-boot-sandal (figures 2.4 and 3.4), CDESA behaves very similarly to ESA. In both Kasparov-Chess (figure 2.2) and Israel-Hebrew (figure 2.1) CHESA is unstable with respect to representation size (similarly to its behavior in figure 2.4) and finally converges to CDESA (the full representations of CDESA and CHESA are identical). For the multiple words tests from figure 2.5, ESA, CHESA and CDESA behave very similarly.

From the examples discussed here it seems that CDESA performs at least as the better of the ESA and CHESA algorithms - it is as order-preserving as ESA (because it selects the most highly associated concepts for any representation size), more consistent than ESA for large representation sizes (because it adds categories to the representation, which smooth the noise of lowly associated articles) and is more stable than CHESA with respect to representation size (because it is not as limited as CHESA in selecting concepts only from the partial hierarchy).

It remains to be seen if the algorithmic differences of CDESA compared to ESA and CHESA have a positive contribution to its performance in the more comprehensive dataset tests (chapter 4), just as ESA and CHESA behave similarly for the above mentioned multi-word examples but achieve different performance scores in dataset tests.
Figure 3.3: CDESA and ESA graphs for cosine similarity and common concepts vs representation size, for the words Kasparov and Chess
Figure 3.4: CDESA and CHESA graphs for cosine similarity and common concepts vs representation size, for the pairs shoe-sock, shoe-boot and shoe-sandal.
Chapter 4

Empirical Evaluation

The examples in chapter 3.5 provide a qualitative assessment of CDESA vs ESA and CHESA. For a more objective evaluation of CDESA, we need to examine it with well established tests - datasets of word or text pairs for which there exists relatedness judgments to which we can compare CDESA’s results.

4.1 Experimental Methodology

4.1.1 Experimental Setup

In order to evaluate the performance of CDESA, we follow a common method used for evaluating relatedness algorithms (used by Gabrilovich and Markovitch (2007) for ESA, Liberman and Markovitch (2009) for CHESA, Budanitsky and Hirst (2006), Gracia and Mena (2008), and others; also see a meta-study of such methods in (Cramer, 2008)) and use the following setup:

- **Dataset:** We use datasets that contain pairs of texts (single or multiple words). Each pair was given to several human evaluators who assigned it a relatedness score. The scores are averaged (to compensate for their diversity which stems from the subjective nature of human evaluations and from the difficulty of humans to give exact numerical relatedness scores rather than rank of relatedness with respect to other pairs) and the average is called the *human judgment* score of relatedness between the paired texts.

- **Computing relatedness by CDESA:** We use CDESA to represent the pairs of texts from the dataset, and compute cosine similarity between them. These similarity measures are taken as the relatedness scores CDESA assigns to the pairs.

- **Agreement with human judgment:** We use the Spearman and Pearson correlation measures to calculate how well does using CDESA for representation give relatedness scores that agree with the human judgment.
We take high correlation with human judgment as our goal. The reason is that although humans may not see or understand all the aspects of a certain text, thus possibly impairing their judgment of its relations with other texts, relatedness is near impossible to measure objectively and is best defined in terms of how humans grasp it - two texts are semantically related if humans perceive them as such, and semantically unrelated otherwise.

4.1.2 Knowledge Repository

For this work, the knowledge repository used is Wikipedia, following the examples of ESA and CHESA to which this work is being compared primarily.

Wikipedia (http://www.wikipedia.org/) is a mostly textual online user edited encyclopedia. It is a very large collection of articles on many different subjects (animals, cosmology, TV series, etc.). Its articles are created and edited by its users, making it a worldwide collaborative effort of knowledge synthesis and production. It is composed as a hierarchical structure, where sets of related articles are collected in categories (e.g. CAT, DOG and ELEPHANT under ANIMALS), and sets of related categories are collected in higher level categories (e.g. ANIMALS, FUNGI and PLANTS under EUKARYOTES). These categories are also user created, and so represent the semantic relations between the articles and other categories as they are grasped in the minds of humans worldwide.

The version used is the same as used in (Liberman & Markovitch, 2009), in order to allow a direct comparison of the algorithm to ESA and CHESA with no change of the underlying data. This is important as Gabrilovich (2006) showed that using a newer version of Wikipedia might change the results, increasing correlations with human judgments. Since Wikipedia is growing at a very fast rate, both by adding new articles and categories, and by expanding existing ones, we expect newer versions of Wikipedia to improve our results (however we did not test this hypothesis). ESA results for WS353 and LEE50 shown here were achieved using a 2006 snapshot of Wikipedia, so they might be improved a little if using our snapshot of Wikipedia (Gabrilovich (2006) reports an improvement of 0.01 in correlation for both WS353 and LEE50 for the Wikipedia snapshot of 2006 relative to a 2005 snapshot). However ESA's results on MTurk were achieved using a more updated version of Wikipedia than our own, and so can be directly compared to our results.

The Wikipedia version used is a snapshot of Wikipedia taken on October 18th, 2007. We follow the filtering method used by Liberman and Markovitch (2009) and by Gabrilovich (2006) for filtering the raw Wikipedia data: Articles with less than 100 non stop-words and articles with fewer than 5 incoming or outgoing links have been removed from the graph. Lists and stubs are also disregarded. All articles’ texts are cleared of stop-words, so stop-words cannot be represented in this method. All words are used as their respective terms by Porter stemming (Porter, 1980) using the official implementation from http://tartarus.org/~martin/PorterStemmer/. Table 4.1 presents
some statistics of this repository.

4.1.3 Datasets

There are very few datasets of words or texts relatedness using human judgment. Most datasets that are similar in nature use similarity judgments rather than relatedness, so they are incompatible with our algorithm. We chose to evaluate the performance of CDESA on three datasets, two of them well explored (WS353 and LEE50), and the third is relatively new (MTurk).

- **WS353** (Finkelstein et al., 2002): 353 pairs of words, assembled from 434 individual words (not all words are paired). Each pair was judged by 13-16 university graduates fluent in English. Judgment scores were on a 0 to 10 scale (0 being totally unrelated, 10 being extremely related or identical), and were averaged to give a single representative relatedness judgment for each pair.

- **MTurk** (Radinsky, Agichtein, Gabrilovich, & Markovitch, 2011): 287 pairs of words, assembled from 486 individual words (not all words are paired). Each pair was judged on a scale of 0 to 5 by an average of 23 Amazon Mechanical Turk workers. Each of the workers has also judged a sample of the WS353 set, and the judgments of those who were not correlated enough with the sample set were discarded. The remaining judgments were averaged to give a single representative relatedness judgments for each pair.

- **LEE50** (Lee, Pincombe, & Welsh, 2005): 1225 pairs of texts, assembled from 50 individual texts (all texts are paired). The texts are short news excerpts, in lengths of 51 to 126 words long, from the Australian Broadcasting Association’s
news mail service. They were judged on a scale of 0 to 5 by 83 Australian university students, with each pair judged 8-12 times. The judgments for each pair were averaged and rounded to 2 decimal places to give a single representative relatedness judgment for that pair.

4.1.4 Correlation Measures

As noted before in section 4.1.1, absolute human judgment scores are not the best tool for the evaluation of how related a text pair is, but rather its rank of relatedness among other pairs is to be used. The Spearman rank correlation coefficient (Spearman, 1904)\(^1\) finds the rank correlation of two sets of observations, and so it is used for the word relatedness tests WS353 and MTurk. However, Spearman’s test is inaccurate when there are many tied ranks. With WS353 and MTurk there is no problem of tied ranks because they provide a pure average of the relatedness scores, and with such small sets there are no two identical scores, so Spearman can be used. In LEE50 the scores were rounded to 2 decimal places, creating only 67 distinct values of relatedness scores out of 1225 pairs. Therefore Spearman cannot be used for LEE50, and we use Pearson’s correlation coefficient\(^2\) instead, that computes the correlation between the actual observations values, i.e. the human judgments and CDESA’s judgments, rather than between their ranks.

The above reasoning is the same as was done in previous works (including ESA and CHESA) so correlation scores can be directly compared between them and CDESA.

4.2 Datasets Experiments and Results

In this section we investigate the behavior of CDESA on each of the three previously mentioned datasets, determining the best \(k\) and \(\delta\) parameters for each. We then assess whether \(\delta\) is a significant factor in achieving good performance and how performance is affected by variations of the algorithm’s parameters.

4.2.1 Determining the Best Parameters for Each Dataset

CDESA has two major parameters: The \(k\) concepts with the highest association score to include in the representation, and the minimal implicit concept coverage that constitutes a new subject, \(\delta\).

We explore how \(k\) and \(\delta\) affect the performance of CDESA. The performance of CDESA is measured as its correlation to human judgment.

Figure 4.1 shows the effect of \(k\) and \(\delta\) on the performance of CDESA for the WS353 dataset. The x-axis is the \(\delta\) parameter. Each line corresponds with a certain \(k\) value

\(^1\rho_{X,Y} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2(y_i - \bar{y})^2}}\) where \(x_i\) and \(y_i\) are the ranks of \(X_i\) and \(Y_i\) respectively; tied ranks receive the average of their positions in the sorted variable

\(^2\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}\)
specified next to it \((k = \infty)\) means full representation). The y-axis is the Spearman correlation with human judgments. The graph was divided to two parts for better readability.

This test’s results reveal some interesting patterns:

- For \(k \geq 2500\), \(\delta\) has a negligible effect. This is because of the correlation presented in section 3.5.1 between association score and \(\Delta_{CAA}\): \(\Delta_{CAA}\) drops as the association score drops. Adding many top associated concepts in advance leaves very little room for concepts with \(\Delta_{CAA} \geq \delta\), so when \(k\) is high, \(\delta\) has a negligible effect on the representation.

- For \(k > 2500\), performance drops. This happens because of the introduction of many less relevant concepts to the representation, which act as noise that masks the more relevant concepts.

- For \(k \leq 1000\), \(\delta\) affects performance, and for smaller \(k\) this effect is more pronounced. This is because for lower \(k\) values, more highly associated concepts are candidates to be added using the \(\Delta_{CAA}\) method, and because of the positive correlation of association scores and \(\Delta_{CAA}\) there are more concepts with \(\Delta_{CAA} \geq \delta\), and so \(\delta\) has a larger effect on which concepts are included in the representations and thus on the algorithm’s performance.

- For very small \(k\) values and high \(\delta\) values, the performance is the worst. This happens for two reasons: Firstly, the low \(k\) value causes CDESA to include only a very small number of initial concepts, and because of the sparseness of the representations common concepts between word pairs are few. This was demonstrated in the Ireland-Dublin example in section 3.5.2. Secondly, the high \(\delta\) value includes only concepts with high \(\Delta_{CAA}\), which means that the representation is implicitly covered very fast, and so very few concepts are added in total.

With a grid search of \(k \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, \infty\}\) and \(\delta \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\), best performance of 0.762 was achieved with \(k = 2000, \delta = 3\). For comparison, CHESA achieved peak performance of 0.72. It did so with full representations, but CDESA achieves its peak performance with a partial representation. As for words CDESA’s full representations are identical to CHESA’s (all articles and categories are selected, and with using the same knowledge repository they are the same; association scores computed identically) this shows that concept selection is beneficial to performance. ESA achieved 0.75, also with full representation; this shows that adding categories to the representation and performing concept selection is beneficial to performance.

Figure 4.2 shows a similar behavior of the algorithm for MTurk. With MTurk, CDESA reaches peak performance with \(k = 50, \delta = 2\) with 0.638 correlation to human
judgments, but very similar performance is reached with most $k$ values and $\delta \in \{2,3\}$. CHESA was not directly tested on MTurk, but since CDESA with $k = \infty$ is identical to CHESA with full representation, we can use this score to say that CHESA achieves 0.62. Again, we see the benefit of concept selection for performance. ESA achieved only 0.59, so with both CDESA and CHESA achieving better performance, this is a testimony to the benefit of abstraction through the inclusion of categories in representations.

Figure 4.3 shows a similar behavior of the algorithm for LEE50. With LEE50, CDESA reaches peak performance with several $k$ values for $\delta = 4$: $k = 750, 1000, 1250$, with 0.739 correlation to human judgments. CHESA achieved peak performance of 0.70 and ESA of 0.72, both with full representations. So again the inclusion of categories and concept selection proves to be beneficial for performance.

Table 4.2 summarizes these results.

### 4.2.2 The Effect of $\delta$ on Performance

Figures 4.4, 4.5 and 4.6 show figures 4.1, 4.2 and 4.3 (respectively) from a different perspective: For each value of $k$, the maximum performance and the standard deviation of performance over the different values of $\delta$ are shown. Having a larger than zero standard deviation indicates that the value of $\delta$ played a role in achieving the best performance for that value of $k$.

The significance of this role is indicated by the magnitude of the standard deviation - a large standard deviation of performance for a certain $k$ value indicates that setting the correct $\delta$ value had a large effect on performance for that $k$.

Obviously, for lower values of $k$, $\delta$ plays a bigger role in determining performance, since it has a bigger role in deciding which concepts would be included in the representation. But $\delta$ affects performance for almost all values of $k$, and most importantly, the peak performance is achieved while $\delta$ affects performance (for MTurk and LEE50 the effect is significant while for WS353 it is negligible).

From this we conclude that $\delta$ is important for achieving good performance on all but the largest $k$ values.

<table>
<thead>
<tr>
<th></th>
<th>LEE50</th>
<th>WS353</th>
<th>MTurk</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>0.72</td>
<td>0.75</td>
<td>0.59</td>
</tr>
<tr>
<td>CHESA</td>
<td>0.70</td>
<td>0.72</td>
<td>0.62*</td>
</tr>
<tr>
<td>CDESA (grid)</td>
<td>0.74</td>
<td>0.76</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 4.2: Best results for datasets (grid search)
* CHESA/MTurk calculated from CDESA with $k = \infty$
4.2.3 CDESA Robustness with Respect to \( k \) and \( \delta \)

We now test how robust CDESA is with respect to \( k \) and \( \delta \) - are there only a few sets of \( (k, \delta) \) values that give good performance, or are there many?

Figure 4.7 shows the difference between the maximum performance of CDESA and the percentiles of the performance of \( (k, \delta) \) in the grid search for the three datasets. The dotted line signifies a difference of 0.01 from the maximum. Since the grid search has 280 data points, each 10 percentile represents 28 \( (k, \delta) \) pairs. This figure shows that for all datasets, about 20% of the \( (k, \delta) \) pairs from the grid search achieved performance that is within 0.01 from the maximum performance. This means that the algorithm is quite robust in that it is not very dependent on specific \( (k, \delta) \) values to achieve peak performance, but that there are many such pairs that achieve very good performance for all three datasets.

4.3 Datasets Results Correlation

Section 4.2 showed results for a grid search of CDESA’s parameters. This provides an insight to how the algorithm behaves with different parameters for different datasets. The best combination of \( k \) and \( \delta \) for each dataset provides an upper boundary (almost - the search was not exhaustive) for the performance of CDESA on that dataset. For a realistic assessment of CDESA’s strength this is not enough. From the above mentioned results we cannot assess whether CDESA may have some specific combinations of parameters that provide good results for all datasets (not only the three we experiment with in this work), or that it can only be fine-tuned to each dataset separately. The latter would be especially problematic if human judgment is unavailable (e.g. a completely new dataset is constructed, or the dataset is huge so calculating human judgments for a representative sample is costly).

4.3.1 Examining Datasets Results Correlation

In this test we explore the correlation between the performance of CDESA on the three datasets with respect to \( (k, \delta) \) pairs. This is done as a preliminary to learning on datasets, to substantiate the hypothesis that such correlations exist and therefore learning is possible.

Figure 4.8 shows the correlation between the results of WS353 and MTurk. Each point represents a certain \( (k, \delta) \) parameters pair from the grid search, the x-axis is the correlation of WS353 with the human judgment for these parameters, the y-axis is the correlation of MTurk with the human judgment for these parameters. One can see that high WS353 results correspond with high MTurk results, and similarly for low results. The correlation between the WS353 and MTurk results is 0.76, a high correlation, meaning that parameter pairs that give good performance in WS353 would give good performance in MTurk and vice versa.
Figures 4.9 and 4.10 show correlation charts between WS353 and LEE50 and between MTurk and LEE50 respectively. Here we see lower correlations between the datasets - between WS353 and LEE50 the correlation is 0.45 and between MTurk and LEE50 it is 0.00. This is to be expected, as LEE50 is vastly different from both WS353 and MTurk - its representations are based on multi-word rather than on single-word texts, and its agreement with human judgments is calculated using Pearson’s correlation coefficient rather than Spearman’s, which are incomparable.

As a curiosity, if Spearman is used to compute agreement scores for LEE50 (although not very suitable for this dataset) then LEE50 Spearman scores correlation with WS353 Spearman scores is 0.78, and LEE50 correlation with MTurk is 0.56, confirming that the type of the correlation coefficient used to calculate agreement with human judges has a considerable influence on the correlations between datasets results; see figure 4.11. Also, although in general Spearman and Pearson coefficients are not correlated, in the case of the LEE50 dataset they mostly are, except for a few outliers, see figure 4.12. This, together with the Spearman correlation between LEE50 and the other datasets, gives further empirical strength to the assumption that the correlation of good parameters between WS353\textsubscript{Spearman} and LEE50\textsubscript{Pearson} and between MTurk\textsubscript{Spearman} and LEE50\textsubscript{Pearson} does indeed exist.

4.3.2 Learning Parameters for CDESA

One way to test for CDESA’s generalization ability is to follow the experimentation methods of machine learning, and to partition a dataset into train and test sets, to learn good parameters on the train set and to apply them on the test set, thus testing the generalization of CDESA from a dataset with a known target to a previously unseen dataset. However, since none of our three datasets have a standard train/test partitioning, we would not be able to compare CDESA’s results directly to other algorithms. We will therefore use one or two datasets as a train set and test the learned parameters on the remaining dataset(s).

Learning from One Dataset

We begin with taking one dataset as a train set and applying the learned good combination of \( k \) and \( \delta \) on the other two. As mentioned in the section 4.2, the best parameter \( \langle k, \delta \rangle \) pairs for WS353, MTurk were \( \langle 2000, 3 \rangle \), \( \langle 50, 2 \rangle \), respectively, and for LEE50 \( \langle 750, 4 \rangle \), \( \langle 1000, 4 \rangle \) and \( \langle 1250, 4 \rangle \) all achieved the same performance. Table 4.3 summarizes these results. For each \( \langle k, \delta \rangle \) combination learned on one dataset, the results of applying it to the two other datasets are displayed.

Note that for all datasets the decline from the upper bound achieved with grid search to results achieved using learned parameters on other datasets was not large. This is in part thanks to the robustness of CDESA with respect to \( k \) and \( \delta \) as demonstrated in section 4.2.3, and in part due to the general correlation between CDESA’s results on
datasets using the same parameters as investigated in section 4.3.1.

WS353 and MTurk show good performance when learning on each other, which is to be expected as they are both single word datasets so their representations arrive from the same distribution, and they both use Spearman’s rank correlation coefficient to calculate agreement with human judges, so their results are comparable.

On the other hand, the largest drops in performance involve LEE50 and MTurk. This can be explained as LEE50 is a multiple-word dataset and MTurk a single-word dataset, so their representations’ distribution is different. Also, LEE50 measures performance using Pearson’s correlation coefficient while MTurk uses Spearman’s. These correlation coefficients are generally not correlated, which can explain why top performance in one does not predict well top performance in the other. These differences do not explain how LEE50 and WS353 give relatively good results for each other. See section 4.3.1 for further examination of this issue.

Learning from Two Datasets

While the results from learning from one dataset are not bad, they are not particularly good. There is still some room for improvement, which we attempt to achieve by learning from pairs of datasets in conjunction. For a given pair of datasets, we take the correlation results from the grid search, and combine them on each \( \langle k, \delta \rangle \) value pair by a combination function (either mean or \( L_2 \) norm). We then find the \( \langle k, \delta \rangle \) pair that gives the maximal value for the combination function over all data points in the grid. We then apply the found \( \langle k, \delta \rangle \) pair to the third dataset and report the results in table 4.4.

Interestingly, for all dataset pairs (WS353+MTurk, WS353+LEE50, LEE50+MTurk), the selected \( \langle k, \delta \rangle \) pair is \( \langle k = 1000, \delta = 3 \rangle \). Although this pair is in the vicinity of the maximal results for each individual dataset, it is not the pair that shows the maximal result for any of them, even if by only a small margin. These results were achieved with both combination functions we tested.

The results of this experiment are very close to the upper bound calculated by the grid search - using \( \langle k = 1000, \delta = 3 \rangle \) for LEE50 gives a performance of 0.737 (versus the 0.739 upper bound), 0.756 with WS353 (vs 0.762) and 0.634 with MTurk (vs 0.638).

The results are surprising in that the learning process gives excellent results despite the relatively low correlation between the word datasets and LEE50 as demonstrated
Table 4.4: Results of learning from two datasets

<table>
<thead>
<tr>
<th></th>
<th>LEE50</th>
<th>WS353</th>
<th>MTurk</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDESA (learn on WS353+MTurk)</td>
<td>0.74</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CDESA (learn on MTurk+LEE50)</td>
<td>—</td>
<td>0.76</td>
<td>—</td>
</tr>
<tr>
<td>CDESA (learn on WS353+LEE50)</td>
<td>—</td>
<td>—</td>
<td>0.63</td>
</tr>
</tbody>
</table>

in section 4.3.1, both for learning from two word datasets and applying on a text dataset, and for learning on a combination of single and multi word datasets and applying on a word dataset. Our conclusion is that learning from a combination of two datasets smooths out outliers and reveals the underlying correlation between datasets as represented by CDESA. Table 4.5 summarizes the learning results concisely.

We conclude by expressing our hope that although tested on only three datasets, the correlations between them and the existence of a single \(\langle k, \delta \rangle\) pair that gives near optimal results for all three datasets suggest that the selected pair \(\langle k = 1000, \delta = 3 \rangle\) would give good results for datasets with limited or no human judgment data. In any case, the method of learning on several datasets should prove useful when new datasets with human judgment data are introduced.

Table 4.5: Results of learning from datasets - summary

<table>
<thead>
<tr>
<th></th>
<th>LEE50</th>
<th>WS353</th>
<th>MTurk</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>0.72</td>
<td>0.75</td>
<td>0.59</td>
</tr>
<tr>
<td>CHESA</td>
<td>0.70</td>
<td>0.72</td>
<td>0.62*</td>
</tr>
<tr>
<td>CDESA (grid)</td>
<td>0.74</td>
<td>0.76</td>
<td>0.64</td>
</tr>
<tr>
<td>CDESA (learn on single dataset)</td>
<td>0.70-0.73</td>
<td>0.74-0.75</td>
<td>0.60-0.63</td>
</tr>
<tr>
<td>CDESA (learn on two datasets)</td>
<td>0.74</td>
<td>0.76</td>
<td>0.63</td>
</tr>
</tbody>
</table>

* CHESA/MTurk calculated from CDESA with \(k = \infty\)
Figure 4.1: WS353 correlation to human judgment vs \( top - k \) and \( \delta \)
Figure 4.2: MTurk correlation to human judgment vs $top - k$ and $\delta$
Figure 4.3: LEE50 correlation to human judgment vs $\text{top} - k$ and $\delta$
Figure 4.4: CDESA WS353 dependency of performance on $\delta$

Figure 4.5: CDESA MTurk dependency of performance on $\delta$
Figure 4.6: CDESA LEE50 dependency of performance on $\delta$

Figure 4.7: CDESA Robustness with respect to $\langle k, \delta \rangle$
Figure 4.8: Correlation of results between $WS353_{Spearman}$ and $MTurk_{Spearman}$

Figure 4.9: Correlation of results between $WS353_{Spearman}$ and $LEE50_{Pearson}$
Figure 4.10: Correlation of results between MTurk_{Spearman} and LEE50_{Pearson}
Figure 4.11: Correlation of results between $WS353_{Spearman}$, $MTurk_{Spearman}$ and $LEE50_{Spearman}$
Figure 4.12: Visual correlation of Pearson and Spearman in LEE50, WS353, MTurk
Chapter 5

Related Work

Methods for the selection of relevant features from representations for linguistic tasks, used to overcome computational difficulty and noise and to improve generalization, have received a lot of attention in the past. It is a more specific task of the general feature selection problem, an introduction to which is presented by Guyon and Elisseeff (2003). Most of the earlier work focused on simple terms features, usually words within documents, with little to no background knowledge about the relations between terms. Only corpus based or general linguistic (such as stop-words and morphology) knowledge was available to the algorithms, meaning deductions could only be made according to the occurrence and co-occurrence of terms in documents, but semantic data for the features was otherwise absent from the calculations. Examples, mostly classification tasks oriented (and mostly supervised), being: Yang and Pedersen (1997) conduct a comparative study on the merits and drawbacks of several methods of feature selection for document categorization - document frequency (removal of rare features), information gain (uses only features that help category prediction), mutual information (using features that tend to occur in a certain category), $\chi^2$ test (testing for feature-category dependency against a known distribution) and term strength (the prevalence of a feature in a similar document cluster). Dumais, Platt, Heckerman, and Sahami (1998) suggest a method that selects the top-k features using the mutual information measure as a preparatory stage for machine learning algorithms of classification. Aizawa (2003) discusses the TF.IDF (term frequency x inverse document frequency) measure of term specificity (and hence importance) and suggests PWI (probability-weighted amount of information), an improved method for the same task that is better grounded in information theory than TF.IDF and provides a better measure of the information a feature carries. Decadt, Hoste, Daelemans, and Bosch (2004) use a genetic algorithm for joint feature selection and parameter optimization, for representing words for a word sense disambiguation task. Baker and McCallum (1998) use distributional clustering of words for aggressive dimensionality reduction for text classification. Mihalcea and Tarau (2004) suggest the TextRank algorithm, that selects the top-k concepts of a representation according to their ‘PageRank’ equivalent in a co-occurrence graph of
Some algorithms do not use the original features of the problem, but rather convert them into another feature space (usually unintelligible to humans) where their representation can be reduced in dimensions with minimal loss of information: Li and Jain (1998) use Principal Component Analysis (PCA) for dimensionality reduction for text classification. Shima, Todoriki, and Suzuki (2004) use Latent Semantic Indexing (LSI), based on a Singular Value Decomposition (SVD) of the term-document matrix, for dimensionality reduction for text classification.

With the development of knowledge databases such as WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) and Wikipedia (http://www.wikipedia.org/), features could have data associated with them, that stems not from the processed documents themselves but from those world knowledge databases. Features were now independent concepts, carrying meaning and inter-concept relations. This could be used for feature generation, as described e.g. by Hotho, Staab, and Stumme (2003) who use WordNet to augment or replace textual terms by corresponding WordNet concepts for document clustering; Mladenić and Grobelnik (2004) who map documents onto the Yahoo hierarchy of web documents for text categorization; and Gabrilovich and Markovitch (2007) who use Wikipedia and the Open Directory Project (ODP) to augment a bag-of-words representation of documents with the corresponding concepts that contain terms from these documents for text categorization, forming the Explicit Semantic Analysis (ESA) algorithm discussed extensively in this work.

Using background knowledge, new algorithms could be developed with the ability to filter features using linguistic and semantic relations that were otherwise unavailable to previous algorithms. While applicable to supervised algorithms, e.g. Gabrilovich (2006) attempted top-k feature selection for ESA, sorted by information gain, with mixed results; Egozi et al. (2008) use a pseudo-relevance feedback algorithm for feature selection for information retrieval tasks, with generated features from Wikipedia (using ESA); this also allows the use of unsupervised methods, that depend not on a specific problem but on the stationary world knowledge, eliminating the need for time consuming algorithm training over training sets that are time consuming to build, and removing the dependency of representations on documents classes. Examples: Liberman (2010) compares ESA’s performance with varying representation size, taking the top-k features ordered in descending document-concept association score. Grineva, Grinev, and Lizorkin (2009) map texts to Wikipedia concepts, and then perform a communities finding algorithm on the semantic graph of retrieved concepts (edges are weighed according to link analysis based concept similarity measures) to select those concepts that are most relevant to the represented texts.

Methods for the unsupervised extraction of relevant concepts from hierarchies using an analysis of the hierarchical structure are not common; most concept selection meth-
ods employ non-structural statistical approaches such as TF.IDF, machine learning, etc. However, following are a few examples of the structural approaches:

In CHESA (Liberman & Markovitch, 2009), documents are represented as sets of concepts from Wikipedia’s hierarchy. The concepts are selected for the representation from a truncated version of the hierarchy of a size given as a parameter to the algorithm, creating a more abstract representation of the document for lower representation sizes. The merits and drawbacks of this algorithm have been extensively discussed in this work.

Similar to our approach of selecting representative concepts from subgraphs of the concept hierarchy, Chen, Xue, and Yu (2008) propose a keyword suggestion method for input queries based on concept selection from an agglomerative clustering of the Open Directory Project (ODP) hierarchy taking into account the coverage of the selected concepts with respect to the query (measured as a variant of tf.idf). The distance metric is based on the frequency of the query in the various concepts. A representative concept from each cluster (the least common subsumer) is then tested for relevancy to the query, and the most relevant concept is used to suggest keywords for the input query. As is, this method does not produce concept vectors; but it can be used to create such vectors if stopped mid-way and the selected concepts with their similarity-to-query scores could be used as the concept vector. Contrary to our approach, this method discards virtually all leaf nodes (taking only the least common subsumers, i.e. inner nodes), ignoring potential relevant information; and also takes just one representative concept from each cluster even if there are other concepts in that cluster that are very similar to the input query, which gives all clusters the same weight in constructing a concept vector, over-preferring diversity over coverage, which was shown in this work to be harmful to the performance of textual relatedness tasks (consider CDESA with low $k$ and high $\delta$ values).

Linked data-based concept recommendation (Damljanovic, Stankovic, & Laublet, 2012) is a method for discovering relevant concepts from the Linked Open Data (LOD) project. The paper suggests two iterative methods for discovering such concepts starting from a user’s query defined as an initial set of concepts. The first is by iteratively adding new concepts based on distance measures over the input hierarchy (enhanced with edges created using concept-relatedness functions), the second by an iterative approximation to LSA to construct a feature vector and then extracting concepts that map to a similar feature vector using cosine similarity. Both methods emphasize relevant concepts discovery and neglect coverage of the hierarchy altogether. While this method fits the problem stated in the paper (discovery of relevant concepts from different fields of knowledge), using it for text relatedness may produce inferior results as diversity and coverage were shown in this work to play a major role in the performance of such tasks.
Several algorithms were tested on datasets used in this work; following are a few prominent examples:

- **IntelliZap** (Finkelstein et al., 2002) uses an external corpus to obtain word-domain vectors, in which each dimension describes the relevance of the word to a specific domain. Word pair relatedness is then determined by the correlation between the corresponding word-domain vectors of both words. It achieved a Spearman correlation of 0.55 for WS353, which this article introduced.

- **Lee et al. (2005)** introduced the LEE50 dataset in the paper *An empirical evaluation of models of text document similarity*. The algorithm described there uses Latent Semantic Analysis (LSA) to train on 314 document from the same news source the LEE50 dataset was composed of, to represent the 50 dataset documents in the resulting latent concept space. Cosine similarity was used to test the relatedness of the 50 documents to each other. A 0.60 Pearson correlation was achieved using this method. Also tested in this paper are count based methods (single word, and 3-gram to 10-gram), but they all achieved correlation coefficients of less than 0.50.

- **WikiRelate!** (Strube & Ponzetto, 2006) maps texts into conceptual representations using Wikipedia’s articles and categories and then measures their relatedness using concept edge distance in Wikipedia, least common subsumer concept and textual overlap between concepts. It achieves 0.55 Pearson correlation on the WS353 dataset using these measures.

- **WikiWalk** (Yeh, Ramage, Manning, Agirre, & Soroa, 2009) uses random walks on the Wikipedia graph to generate a PageRank evaluation of nodes according to their relevance to a text. It uses an ESA representation as a starting vector for the random walk. The representation is trimmed to its 625 most highly associated concepts (625 determined by grid search on LEE50), and the walk is performed only on a subset of Wikipedia’s categories, also determined by grid search. It achieves 0.77 Pearson correlation on the LEE50 dataset. However, the heavy influence of the grid search on performance (e.g. with full ESA representations as starting vector the performance is only 0.71) and the fact it was only tested on one dataset on which the grid search was also performed, suggests that it should be further tested on other datasets for generality.

- **Temporal Semantic Analysis (TSA)** (Radinsky et al., 2011), another conceptual representation algorithm, uses a temporal database (such as a newspaper; in this paper the New York Times was used) to extend ESA’s representations with temporal data. It does so by assigning each concept in the representation its time series of occurrences in the temporal database and then compares representations according to the similarity of their corresponding time series. It achieves 0.80
Spearman correlation on the WS353 dataset, and 0.63 on the MTurk dataset, which this paper introduced.
Chapter 6

Conclusions

Conceptual representations of texts are a means to semantically represent texts using human-created repositories of knowledge. A concept is an abstract representative of a portion of the knowledge repository. For representing a text, each concept is given an association score to that text. The set of \( \langle \text{concept}, \text{association score} \rangle \) pairs constitutes the conceptual representation of the text.

Texts can be tested for relatedness by comparing their conceptual representations, where each concept’s association score is compared between the texts, and an aggregate score is given as the relatedness of the two representations.

Including many concepts from the knowledge repository in all representations increases the coverage of information but gives rise to the problem of noise - many less relevant concepts with low association scores may bias the similarity score of two representations, therefore giving false high or low relatedness values to texts. Avoiding noise by selecting only the top associated concepts for each representation presents a new problem - the sparseness of conceptual representations causes very few concepts to be present in both representations, reducing the validity of the calculated relatedness score.

In this work we proposed an algorithm, CDESA, that addresses both noise and coverage problems by refining conceptual representations so that they include concepts according to the coverage of relevant concepts principle. Concepts are deemed relevant if they have a very high association score or if they add to the semantic diversity of the representation. The semantic diversity of a representation is measured with respect to coverage of abstractions of concepts in the representation, using a knowledge repository that is hierarchical; an abstraction of a concept is an ancestor node of the concept in the hierarchy.

We have shown that this algorithm has better performance than previous methods of conceptual representations, ESA and CHESA, when using the same knowledge repository, and that our algorithm produces representations that are more consistent than ESA’s and CHESA’s in conveying the semantics of the represented texts.
For future work we would like to examine CDESA for tasks that can utilize semantic relatedness detection. As a metric to measure distance between texts, it can be used for text categorization, word sense disambiguation and other related tasks. The algorithm’s performance demonstrated in this work shows that the coverage of relevant concepts principle is sound and that the correlation of performance for the datasets we have in hand is promising, and so we believe that this algorithm can be used for the benefit of the very many tasks that require the judgment of semantic relatedness between texts.
References


We aim to describe various meaningful topics, despite it not being a direct representation of the text in a straightforward way.

Inclusion of the concepts is done iteratively, in a hierarchical order by their weight, and by checking the number of ancestors of the concept in question.

If the number of ancestors is greater than the number in the external parameter of concepts that are not ancestors of concepts that are already in representation, then it is added to the representation.

In this way, the representation that includes concepts that describe the text completely includes interpretations of the main concepts of weights less than the meanings of the relationships.

In the case of relations between CHESA and ESA, the method is more effective than the others. Moreover, the method solves problems of stability, noise and flatness. In addition, despite the fact that the method is parameterized, it reflects the empirical stability of different parameter sets for different problems, so that it is possible to learn the parameters from the tested framework and use their success in new cases.

The entire book is divided into chapters, with various meaningful topics and understanding the subject.

iii
Shehory and colleagues [2011] argue that this observation is due to the fact that the system is not able to correctly identify the relationships between concepts that are not directly mentioned in the text. This is because the system is unable to recognize the semantic relationships between concepts, such as those that are based on synonyms or related terms.

A compact version of the hierarchical ESA method [Shehory et al. 2011] is used to represent the text. In this method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.

In the Compact ESA method, the system uses a compact representation of the text to identify the relationships between concepts. The system uses a set of rules to identify the relationships between concepts, such as those based on synonyms or related terms.
The abstract introduces the topic of natural language processing, specifically focusing on the need for semantic representation of words and texts. The text explains that semantic representations are crucial for tasks such as relevance ranking, document summarization, and text classification. However, statistical approaches to text representation lack some key shortcomings, such as the inability to handle implicit meaning and the lack of adaptability to short texts and individual words.

The text highlights the importance of using external knowledge sources to enrich semantic representations. These sources, such as Cyc, an online encyclopedia, are used to provide additional context and understanding of texts. Explicit Semantic Analysis (ESA) is introduced as a method that utilizes this knowledge to represent words and texts, taking into account both context and frequency of usage.

Despite its effectiveness, the method has limitations, particularly in dealing with the vastness of the knowledge base and the complexity of representing implicit meanings. The text concludes by noting the ongoing research and development in this area, highlighting the challenges and potential solutions.
לחינוי, שבשלושת週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週週 Monad מעורר עלייה.
 כיום

אנני מודד להנחתו של, פרופ' שאול מרקוביץ', על התוכנית על התמקדות של במת שנות למידה התואר המשי של. על רעיונות, יזונים, בקורת בונה על עיר, בלאיזונים עבודה וללא היות באמר
ליד הנסהמה.

ודי רבו על קיניה סרטי וליניות שלברם על שgreso רוב של שרת על כיסיה, או ישות כמיה

ġג מלקסיווועית, על היוגה שג בשמול של חתיהם צרי או מית.

ודי לבר, יוחאי צו, ימי אלבון, שיר ברישראל פת נימיץ, על שגנה על לוחות בבילארד.

לנגה אלגרבל, ענף החבש וסיג בורפועת, על עירוד ורי שכטרפיט ואחר. לטחייתложитьת
לארה, עומר ולי, אר זรห, עפר אוצי, אסף נואר, ואיראל ריב, על הע 이루ות והענותה.

ሌבר מתחים במודש שני למידה, על שערו על twisting יג אוצי במחזור רבכ.

ודי מתחים להברית, יו חכים, שתמיד היוגה היא בשבי, מבינו, חתית, מעדות, יתרה.

יאלי מילוס היוגה צרי שמצאו ומתיי רב להטבוק.

ודי על אר צוremium, שלמה עדית טולוון, על געתיי בכי בשחון אם, על תורות והתמקמות
שאנין וודע גבולה, והתוכנית על, על התמקדות על עיר עירוח ב实施意见ה שלג.

ודי על תודות עם, ענירורים והטרפיט על וי אתר שישב מזווית להפגת מתוח.

אני מודד להנחתו על התמקדות המספיט הנגדה בחשתלומטיה.
זיקוק מונח-ביסיוי של ייצוגים מרשים

היבור על מחקר

לע hå לווולוז הידריוות קבגלה לואור
מקסום למידות בדיעי המחשב

任何形式

הוגי טולידנוב

הוגש לפקנוו הכותג - מכון טכנולוגיה לישראל
אב תשע"ב חיפה יולי 2012
זיקוק מונחה-verige של ייצוגים מורפנחים

חגי טולידנו