Text Comparison Using Soft Cardinality

By

Sergio Jimenez
sgjimenezv@unal.edu.co

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

PhD in Information Systems Engineering and Computer Science

Under the guidance of

Alexander Gelbukh and Fabio A. Gonzalez

Systems and Industrial Engineering Department, Universidad Nacional de Colombia, Bogotá, D.C.

2015

UNIVERSIDAD NACIONAL DE COLOMBIA

(NATIONAL UNIVERSITY OF COLOMBIA)

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Sergio Gonzalo Jimenez Vargas

This thesis has been read by each member of the following graduate committee and has been found to be satisfactory.

____________________	_________________________
Date	Dr. Alexander Gelbukh
____________________	_________________________
Date	Dr. Fabio A. Gonzalez Osorio
____________________	_________________________
Date	Dr. Luis F. Niño
____________________	_________________________
Date	Dr. Grigori Sidorov
____________________	_________________________
Date	Dr. Carlos Mario Zapata

Abstract

Text similarity is the task of comparing words, names, phrases, short texts and documents in a way that the scores obtained by automatic methods may be, to some extent, according to human judgment. The set-based approaches for addressing this task consist on representing texts as sets of elements (e.g. sets of characters, syllables, words, n-grams) and compare them using resemblance coefficients as the Jaccard's index and others. This approach is simple and effective but suffers from several issues: i) lack of notion of order, ii) the inability to model element repetitions weighting of elements, iii) lack of adaptability to the task when there is the availability of training data, and iv) the inability to model similarity between elements. Some of these issues have been addressed in the past, but others remain partially or completely as open questions. In this dissertation, the already addressed issues are placed in a comparative context with other methods for text similarity. For example, the use of n-grams, bag theory and parameterized resemblance coefficients for addressing i), ii) and iii) respectively, are presented and compared with other approaches.

The similarity between elements is exploited in a new approach called soft cardinality, which generalizes the classical cardinality of sets and bags. Soft cardinality allows the construction of similarity functions in combination with known resemblance coefficients but provides new functionality and properties. This approach allows the inference of non-empty intersections between collections that do not share identical elements but contain similar ones. The mathematical properties of soft cardinality are presented and discussed both theoretically and empirically to provide a coherent framework for practical applications.

The limited adaptive capacity of similarity measures based on cardinality is addressed by a new approach that extracts cardinality-based features from the objects being compared and builds similarity functions tailored to a particular task. These cardinality-based features are combined with supervised learning methods to induce similarity functions from training data. The proposed methods not only aim to learn such functions, but also suitable representations for each task.

The empirical effectiveness of the proposed approaches was successfully tested in the challenging evaluation contexts of the SemEval competitions in 2012, 2013 and 2014. The proposed methods were used deal with various text processing tasks (e.g. text similarity, textual entailment, student answer scoring, among others) and the systems proposed by us (and others) were always been among

the top winners. A summary of this experience that describes the used methods and shows results comparatively is presented.

Finally, soft cardinality in combination with the cardinality-based feature extraction and selection methods were used for building a lexical similarity function based on WordNet. This new function renews awareness in knowledge-based approaches to lexical similarity, which have been losing interest because of the recent success of neural word embedding. Our lexical-similarity method obtains competitive results versus neural word embedding approaches reaching state-of-the-art results in several benchmarks.

It is concluded that the soft cardinality and cardinality-based features are an important contribution to the set-based approaches extending its use in text applications from being mere baselines to competitive methods in the state-of-the-art. Also, soft cardinality has the potential of being used in other fields in information technology and different domains as demonstrated by the recent connection made to the field of ecology.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The automatic assessment of text similarity refers to models that compare words, names, phrases, short texts, documents, and provides a quantitative score of their similarity or relatedness. To some extent, these scores should reflect the judgments of similarity made by humans over the same texts. This task has been attempted for over five decades using a variety of approaches and aiming at almost any level of combination of text categories ranging from lexical similarity to retrieval and classification of documents. Usually, the text comparison methods are based in a variety of basic techniques borrowed from set theory, vector spaces, probability theory, information theory, stringology, machine learning, and others. These basic techniques are combined with particular methods and resources of the fields of computational linguistics and natural language processing, as taggers, parsers, lexicons, thesaurus, semantic networks, encyclopedic knowledge, corpora among others. In this endeavor, many methods inspired by text applications (e.g. LDA [?]) have proven effective in addressing problems in other fields. With this in mind, this dissertation aims to contribute at the same time to different facets of the text similarity problem and, in general, to set and bag theories with a new approach wich we call *soft cardinality*. In Chapter 2, a survey of text similarity is presented to provide the proper context for the use of soft cardinality in text applications.

Soft cardinality (2010) [?] is a generalization of classical cardinality in set and bag theories, which exploits similarities between collection's elements to provide a "soft" count of the number of elements. In this dissertation, this new cardinality function is theoretically defined and its main properties are verified both theoretically and empirically. Chapter 3 introduces soft cardinality by presenting an intuitive motivation, a formal derivation, its definition, a review of its properties, and an empirical validation using synthetic data. There, the pseudo-monotonic property of soft cardinality is primarily studied because it differs from the formal mathematical notion of a measure. That is, a formal measure never reduces its magnitude when the measured object grows. The argument in favor of using a non-monotonic measure was illustrated by an example that uses values of lexical similarities obtained from statistics in a large corpus. That example showed that the pseudo-monotonic property of the soft cardinality is useful for modeling the non-monotonic behavior of the compositionality of the language, i.e. more words not always conveys more information (see Section 3.5.2).

Interestingly during the development of this dissertation, Leinster and Cobbold, researchers in the field of ecology (2012) [?] obtained a measure identical to soft cardinality aiming to measure diversity in a community exploiting the similarity between species. Both soft cardinality and their zero-order diversity are equivalent, but were derived using different motivations and paths. While soft cardinality is related to set (1874) [?] and bag theories (2001) [?], zero-order diversity is a result that can be traced back to Jaccard (1901) [?], Dice (1945) [?], Shannon's information theory (1948) [?], its Renyi generalization (1961) [?] and the Hill's numbers (1973) [?]. In the times of Jaccard and Dice, the cardinality of sets and species diversity were a single concept, then diversity measures followed a different path, and now 70 years later they meet again.

Lets consider the following example. When measuring diversity in a community of 100 individuals of various species, if say another 100 individual are being added to the community, its degree of diversity can be increased or decreased depending on the species of the individuals being added. For instance, if the added individuals belong to a single species that is already present in the community, then community diversity decreases because one species could become dominant over others. Differently, if the 100 individuals added are an assortment of species that are not already present in the community, then community diversity increases. This example illustrates the fact that diversity measures are non-monotonic because it is quite intuitive that the addition of individuals to a community does not necessarily implies an increase in its diversity. Soft cardinality and zero-degree diversity extend this idea also to similar species. That is, the addition of individuals with a different degree of similarity with respect to the individuals in a community could whether produce an increase or decrease in diversity. One of the contributions of this dissertation is the observation that this non-monotonic behavior of species in a community can also occur analogously in another context such as words in a text. In addition, this fact provides additional theoretical and empirical support to zero-order diversity, while zero-order diversity reinforces the argument in favor of soft cardinality non-monotonicity by the fact that modern diversity measures are non-monotonic by nature.

The idea of soft cardinality allows the construction of similarity measures in different ways. The natural approach is to use soft cardinality as a replacement for classic cardinality in any of the numerous existing resemblance coefficients (e.g. Jaccard, Dice, Tversky, cosine, overlap, etc.) The only additional requirement for computing soft cardinality is to provide a similarity function that compares pairs of elements. For instance, two short texts A and B can be represented as collections of words. In turn, words can be represented as sequences of letters that can be compared with an off-the-shelf measure such as edit distance [?] or Jaro-Winkler similarity [?]. Thus, having a similarity measure to compare words, the calculation of soft cardinality for each text is straightforward.

The calculation of the soft cardinality of $A \cup B$ is simple too, just merge the two collections and follow the same procedure. However, the calculation of the soft cardinality of the intersection is not obvious if $A \cap B = \emptyset$. For that, the well-known set equation $|A \cap B| = |A| + |B| - |A \cup B|$ can be applied for inferring the soft cardinality of the intersection. Finally, a similarity score for A and B can be obtained by the ratio of $|A \cap B|$ and $|A \cup B|$, i.e. Jaccard's index.

Somehow surprisingly, this simple approach was used in the first of the recent SemEval campaigns in 2012 for addressing the semantic textual similarity task [?] obtaining a third place among 88 participating systems. This system [?] obtained a score of 0.6708 in the official performance measure (mean) using a variation of the Tversky index with three fit parameters, a list of stopwords in English and the Porter's stemmer. Comparatively, the best system [?] scored 0.6773 but included in its list of used resources: dictionaries, distributional thesaurus, monolingual corpora, multilingual corpora, Wikipedia, WordNet, distributional similarity, knowledge-based similarity, lemmatizer, POS tagger, statistical machine translation, string similarity, textual entailment and others. While our system, based on soft cardinality, reproduced its results in seconds, most other approaches required hours for processing all resources used. This successful participation initiated a series of participations (by us and other teams) in the same and several other tasks using soft cardinality as core method. Chapter 4 present a summary of that experience to explain some of the methods to use soft cardinality in natural language processing.

In Chapter 5, the accumulated experience in the use of soft cardinality in text applications was used to address a more challenging endeavor. In the recent years, the so-called *neural word embeddings* [? ?] appeared in the scene of natural language processing showing to be the best approach for addressing lexical similarity [?] and semantic textual similarity [?]. This approach consists in obtaining vector representations of words and texts by training predictive models of the words using large text corpora. The computational resources needed for obtaining such representations are enormous, but the ease of use of the representations and their quality largely pay the cost. The success of this approach has eclipsed traditional methods for lexical similarity based on resources like WordNet. In spite of the linguistic richness of the information contained in WordNet, the lexical similarity functions based on it are not match for neural word embeddings. We propose a new cardinality-based approach for lexical similarity that exploits the WordNet's graph using the techniques described in Chapter 4. Our experiments showed that this approach can produce competitive results (even better in some benchmarks) compared with neural word embedding, especially in lexical similarity. However, in the facet of lexical relatedness, neural word embeddings remains unrivaled.

1.1 Problem definition and research questions

The textual similarity task consist in assigning a similarity score to two pieces of text. The main difficulty is that the assigned score must be according to human

Table 1.1: Examples of textual similarity from the SICK dataset (2014) in a 5 to 1 scale

texts	**similarity**
Someone is banging the lens of a camera against a nail	4.5
The man is hammering a nail with a camera	
Some instruments are being played by a band	3.0
A man is playing an electronic keyboard	
A man is singing a song and playing the guitar	1.2
A man is opening a package that contains headphones	

Table 1.2: Examples of lexical similarity from the SCWS dataset (2014) in a 10 to 0 scale

word #1	**word #2**	**similarity**
war	*battle*	9.08
money	*currency*	7.90
peace	*amity*	6.20
drive	*ride*	4.40
energy	*laboratory*	1.90
field	*handle*	0.00

judgment. Table 1.1 shows examples of pairs of texts labeled with the average of the scores of similarity provided by ten different subjects. This problem is mainly composed of two aspects: cognitive and linguistic. The cognitive component refers to the way humans make an assessment of commonalities and differences between the texts to produce a similarity score. The linguistic component deals with the problem that sometimes the texts that share similar words conveys different meaning and texts with no words in common could mean the same thing.

A sub-problem of the text similarity task is the lexical similarity, which involves assigning a similarity score for pairs of words, again, according to humans. Table 1.2 shows examples of pairs of words labeled with the mean scores of similarity provided by ten different subjects. Similarly to textual similarity, lexical similarity is affected by cognitive and linguistic issues (e.g. rareness and polysemy). It is natural to think that if the scores of similarity between the two texts words are available, then these scores can be combined in some way to produce a similarity score for the two texts. However, one of the issues that arise in this approach is the compositionality of the language. For instance, even though the word pairs *hard-rigid* and *disk-plate* are similar a *hard disk* (a storage device) is quite different from a *rigid plate* (a non-flexible dish).

The concrete practical problem addressed in this dissertation is how to obtain similarity scores for pairs of words and how to combine these scores to obtain similarity scores for pairs of texts. These problems have been addressed in the past using different mathematical models to represent texts and to produce similarity scores. Probably the most studied approach is to represent the texts and words as vectors in a high-dimensional Euclidean space (geometric model) and the distances in that space are associated to dissimilarities among words and texts. The mathematical tools for such spaces are numerous, well-known, and proven in many domains (e.g. distance and similarity functions, dimensionality reduction, etc.).

Amos Tversky, a renowned mathematician-psychologist, demonstrated experimentally that the analogy of sets modeled similarity judgments of humans better than the geometric analogy [?]. In addition, the concepts of set, cardinality (number of elements of a set), union and intersection are so simple and intuitive that are introduced from elementary school. By contrast, highly dimensional spaces are not obvious, and even a fourth dimension is difficult to associate with the world that we can perceive with our senses. This could lead us to think that if our goal is to build a model that reproduces human similarity judgments, then a set-based model should be a better option than a geometric model.

Despite this, similarity models based on sets are rather simple and do not have the resources and development available of geometric models. This dissertation aims to answer the question of how can the set-based models be used and enhanced to address textual and lexical similarity tasks. The answer is derived through the proposal, analysis, development and testing of the soft cardinality and the cardinality-based feature representations.

1.2 Objectives

The main goals of this dissertation are:

- propose a new set-based model of similarity, which based on current models, integrates the similarities between elements as a key factor for addressing the textual and lexical similarity tasks.
- provide method for making the proposed model adaptable to training data and capable of combining evidence of different nature.

The main contribution of this dissertation consist in the proposed methods that achieved these objectives. This achievement is demonstrated in a number of theoretical arguments, experiments and comparisons with other approaches. An incidental, but even more important contribution, is the result that the methods proposed for the particular text applications addressed, have the potential to be applied in different domains and scenarios.

1.3 Dissertation road map

The chapters of this dissertation are conceived as articles that can be read independently and in any order. Most of the cross-references are circumscribed within each chapter, except Chapter 2 (background) that links related work to some parts of the dissertation. Inevitably, there is some overlap we tried to minimize while maintaining the consistency of each chapter.

Chapter 2 is intended to provide an introduction for readers with little or no prior knowledge of text comparison problems. Readers with some background in the area could skip sections or the entire chapter.

Chapter 3 provides a complete theoretical definition of soft cardinality and is aimed at readers interested in a profound vision of the proposed model. This chapter should be read by researchers seeking connections with other theoretical models or professionals who want to understand the properties of soft cardinality that would be useful to model different phenomena. This chapter may be skipped by readers who only want to know how to use soft cardinality and want to try it for NLP tasks.

Chapter 4 provides a concise definition of soft cardinality and describes the methods proposed to use it to tackle NLP tasks. In addition, this chapter includes a review of the systems using soft cardinality in recent SemEval competitions. All tasks addressed involve automatic analysis of short sentences and paragraphs from video descriptions, definitions, headlines, short news, questions, answers, and others, which are written in languages such as English, Spanish, French, German and Portuguese.

Chapter 5 addresses the problem of semantic lexical similarity in English using soft cardinality and cardinality-based representations having WordNet as a primary resource. The proposed method is compared with the state of the art using of 12 benchmarks for lexical similarity and relatedness.

Finally, in Chapter 6 we provide the key findings that can be derived from this dissertation, along with a practical section “takeaways”. In addition, we present an inventory of publications produced by the time this research was conducted and some statistics of citations up to date.

Chapter 2

Background on Text Similarity

Text similarity relates to methods for comparing pairs of texts to produce automatically a similarity score. This score must follow the human judgment. In fact, the average of the judgments of various human scorers, and their degree of agreement are the current upper bound for the task, which is commonly used as gold standard for supervised learning methods and evaluation. In our context, "text" could mean any piece of written information ranging its textual level from a word to a document. Our goal in this article is to present the basic underlying approaches to tackle this task. Most approaches are general methods that can be applied, to some extent, to different textual levels ranging from lexical to document similarity and several combinations in between. The presentation of these methods is organized by their underlying modeling assumptions and some comments about the pros and cons of their use at different textual levels are discussed in each section.

A version of this chapter was submitted to *Computación y Sistemas* journal.

2.1 Introduction

2.1.1 Lexical similarity

A key part of text similarity is the comparison of the lexical units that compose texts, i.e. words or multi-words. The objective is to provide a numerical assessment of the similarity of two lexical units as an intermediate resource for the main goal of comparing a pair of texts. The role of lexical similarity in text comparison ranges from pre-processing tasks (e.g. spell-checking, stemming, named-entity recognition) to the identification and quantification of the semantic relations between words. Lexical similarity have two principal modalities; morphologic (words are compared by their letters) and semantic (words are compared by their meanings.)

2.1.1.1 Morphologic lexical similarity

Lexical units are the smallest unit that conveys a meaning in a language. In most of the Indo-European languages, smaller subdivisions of the words into letters, phonemes or syllables are meaningless. Some prefixes and suffixes convey semantic information by changing of meaning of the base word (e.g. *ex*president, police*man*), while others provide grammatical inflexions of the root or stem of the word (e.g. sing*ing*, common*ly*). A side of the syntactic and semantic information provided by these word components, the comparison between words based on morphological information do not provide semantic relationships beyond words that share the same root (e.g. *beauty* and *beautiful*). Although, these semantic relationships based on morphologic similarity are a relatively easy-to-use resource, many false relationships can be inferred from morphologically-similar but semantically-distant words (e.g. *trail* and *trait*.) In spite of that, morphologic lexical similarity generally provides a good baseline for semantic text applications [**?**] and is a useful tool when the texts are noisy due to misspellings, typos, speech recognition and optical character recognitions (OCR) errors. Some of the techniques used for morphologic lexical similarity came from the field of stringology (see [**?**] for a survey.)

The morphological similarity is particularly useful for comparing proper nouns. Names generally lack of a particular meaning in the language and their primary function is to represent a particular real world entity. The superficial morphological representation of names is an important resource for their comparison. Moreover, proper nouns are more prone to typos and spellings variations than regular words (see [**?**] for a survey on name matching.)

2.1.1.2 Semantic lexical similarity

Semantic lexical similarity aims to compare words based on their meanings. Unlike morphologic, semantic lexical similarity requires an external source of information to compare words. The main resources used for this purpose are knowledge sources and distributional representations. Commonly used knowledge sources are semantic networks (e.g. WordNet [**? ?**]), dictionaries, thesaurus (e.g. Roget's thesaurus [**? ?**]) and encyclopedias (e.g. Wikipedia [**?**].) Distributional representations are based on the so-called *distributional hypothesis* that states that the meanings of the words can be deduced from the contexts in which they occur if a sufficiently large corpus is available [**?**]. These resources are exploited in different ways to produce a numerical score of similarity between pairs of words (see sections 2.4.3 and 2.5.1 for some examples.)

There are many kinds of semantic relationships between words, e.g. synonymy, antonymy, hypernymy, holonymy, and many more. However, for text comparisons they are simplified in two categories rather blur: similarity and relatedness. Similarity involves pairs of words such as {*car*, *bike*}, and relatedness pairs like {*car*, *speed*}. The semantic lexical similarity methods are evaluated

Table 2.1: Publicly available datasets for lexical similarity and relatedness

Name	**# word pairs**	**Task**	**Reference**
MC	30	similarity	Miller and Charles [?]
RG	65	similarity	Rubinstein and Goodenough [?]
WS353	353	both	Finkelstein et al. [?]
WSS	203	similarity	Agirre et al. [?]
WSR	252	relatedness	Agirre et al. [?]
SCWS	1,997	similarity	Pennington et al. [?]
RW	2,034	similarity	Luong et al. [?]
MEN	3,000	relatedness	Bruni et al. [?]
YP-130	130	similarity	Yang and Powers [?]
MTurk287	287	relatedness	Radinsky et al.[?]
MTurk771	771	relatedness	Halawi et al. (2012) [?]
Rel-122	122	relatedness	Szumlaski et al [?]
SimLex-999	999	similarity	Baker et al. [?]

by measuring the correlation of their results versus datasets annotated either with similarity or relatedness by humans. Table 2.1 shows a list of the current publicly available lexical similarity benchmarks.

2.1.2 Textual similarity

One of the first studies in semantic textual similarity was conducted by Lee et al. [?], who proposed a dataset of 50 documents with graded human judgments of similarity provided for each possible pair. In addition, they tested Tvesky's, Jaccard's, cosine and overlap coefficients, combined with several representations ranging from 3-grams to 10-grams of words. Other considered representations were obtained using LSA [?] with different term weighting mechanisms reaching. The best configuration reached the inter-rater correlation using LSA model with 100 to 150 latent factors and a local term weighting based on $\log(tf)$ (*tf* means term frequency, the number of occurrences of a term in a document.)

Recently, Agirre et al. [? ? ? ?] proposed a series of SemEval challenges for the semantic textual similarity task (STS). During these campaigns, they have proposed more 26 datasets composed of pairs of texts labeled with human judgments of similarity. In 2014, in the context of the same SemEval competitions, Marelli et al. [?] proposed the SICK dataset (sentences involving compositional knowledge) consisting of 5,000 pair of short texts labeled with human judgments of relatedness and entailment. Simultaneously, Jurgens et al. [?] proposed four datasets each one of approximately 1,000 pairs of texts in different textual levels, i.e. paragraph to sentence, sentence to phrase, phrase to word and word to WordNet's senses.

Given the competitive nature of the challenges in SemEval, the participating systems (more than 280 systems in the last four years) used a variety of combinations of methods and resources. Instead of reviewing these combined approaches, this chapter aims to present a broad overview of the basic techniques used by these systems.

2.2 Notation

Lets first define a notation that establish a common framework for comparing and understanding different approaches for text similarity. Let A and B to objects to be compared with the aim of obtaining a quantitative notion of their similarity. In different contexts in this survey, A and B could be pairs of words, multi-words, sentences, short-texts, documents, etc. In fact, we will show that many of the approaches are analogous and applicable at different textual levels.

Now, to compare A against B it is necessary to build a similarity function $SIM(A,B)$ that returns a number that reflects the similarity them. The first step is to provide an object representation. The simplest one is using objects by themselves, which only makes possible binary comparisons e.g. either $SIM(A,B) = 1$ iff $A = B$ or 0 otherwise. If a gradual notion of similarity between A and B is required, then they must be represented with any subdivision of features for each one. Lets represent A and B as collections of such features:

$$\begin{aligned} A &= \{a_1, a_2, \ldots, a_{|A|}\} \\ B &= \{b_1, b_2, \ldots, b_{|B|}\} \\ w(a_i) &\in \mathbb{R} \\ sim(a_i, b_j) &\in \mathbb{R} \end{aligned}$$

This representation is general enough to model vectors, sets, bags, graphs and ordered sequences. For vectors, the sub-indexes on each feature indexes each dimension in the multidimensional space in which A and B are represented, implying that space have dimensionality $k = |A| = |B|$. In addition, the features a_i and b_i are numbers that indicates the representing value of A and B in the i^{th} dimension in such space. The function $w(i)$ provides importance weights to the i-th features. This weighting function can be as simple as $\forall x : w(x) = 1$ or it can have different or more arguments e.g. $w(i)$ or $w(A, a_i)$. The function $sim(i,j)$ aims to model similarity relations between the features. Again, the simplest choice is $sim(x,y) = 1$ iff $x = y$ and 0 otherwise. For instance, if A and B were words they could be represented by vectors in a space indexed by the possible letters; features a_i could encode the number of times each letter occurs in the word A; function $w(i)$ could give more importance to vowels in comparison to consonants; and $sim(i,j)$ could reflect the similarity between letters due to proximity on a keyboard.

2.3 Set-based similarity

The set-based approaches compare pairs of objects subdividing them into elements and counting the number of different and common elements. Typically,

Table 2.2: Derivation of cardinalities when comparing A and B from $|A|, |B|, |A \cap B|$

$\|A \cup B\| = \|A\| + \|B\| - \|A \cap B\|$	$\|A \setminus B\| = \|A\| - \|A \cap B\|$
$\|A \triangle B\| = \|A \cup B\| - \|A \cap B\|$	$\|B \setminus A\| = \|B\| - \|A \cap B\|$

for comparing words, they are subdivided into characters and texts into words. Although, this approach is conceptually simple and empirically effective, it ignores information like element's order and the element repetitions. However, these two issues can be respectively addressed by using n-grams representations (Section 2.3.2) and the theory of bags [?] (Section 2.3.3.)

Let $A = a_1, a_2, ...a_{|A|}$ and $B = b_1, b_2, ...b_{|B|}$ be sets whose main property is their cardinality (the number of elements) denoted as $|A|$ and $|B|$. Their similarity is defined as follows:

$$SIM(A, B) = F(|A|, |B|, |A \cap B|) \tag{2.1}$$

Where F is an algebraic function that combines these three cardinalities. Function F only requires these three arguments because all possible areas in the Venn diagram of two sets can be derived from these three cardinalities (see Table 2.2). Practically, all resemblance coefficients can be expressed using these arguments.

Other formulations consider the universe of discourse U. This component is used to assess similarities using not only common elements between A and B but also the common-missing elements between them. For instance, in a collection of documents where the majority of the documents mention the city of New York, the fact that a particular pair of documents does not mention it could be considered as a common feature. However, in the ad hoc formulation of similarity in this section, where the similarity score depends only on A and B, the commonalities due to missing elements, is not considered. In practice, in text applications the universe of discourse is commonly large, i.e. the vocabulary of the language, collection or domain, making the effect of common-missing elements is negligible.

The following sections provide different alternatives for the function F.

2.3.1 Resemblance coefficients

Resemblance coefficients are rational functions that return a similarity value in $[0, 1]$ range using cardinalities involving two sets. They have been used in practically all areas of knowledge since Jaccard [?] introduced his famous index more than a century ago. A summary of common resemblance coefficients to compare two sets (without making use of the universe of discourse) is shown in Table 2.3.

The first five coefficients in Table 2.3 are well-known off-the-shelf measures. Although, similarity scores provided by these coefficients are highly correlated

Table 2.3: Named Resemblance Coefficients

resemblance coefficient	**expression**
Jaccard [?]	$\frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$
Dice or Sørensen [?]	$\frac{\lvert A\cap B\rvert}{0.5(\lvert A\rvert+\lvert B\rvert)}$
Overlap	$\frac{\lvert A\cap B\rvert}{\min(\lvert A\rvert,\lvert B\rvert)}$
Cosine or Ochiai [?]	$\frac{\lvert A\cap B\rvert}{\sqrt{\lvert A\rvert\cdot\lvert B\rvert}}$
Hamming	$\frac{1}{1+\lvert A\triangle B\rvert}$
Generalized Dice	$\frac{2\cdot\lvert A\cap B\rvert}{(\lvert A\rvert^p+\lvert B\rvert^p)^{1/p}}$
Tversky [?]	$\frac{\lvert A\cap B\rvert}{\alpha\lvert A\setminus B\rvert+\beta\lvert B\setminus A\rvert+\lvert A\cap B\rvert}$
Symmetrized Tversky [?]	$\frac{\lvert A\cap B\rvert}{\beta(\alpha a+(1-\alpha)b)+\lvert A\cap B\rvert}$ †
De-Baets & De-Meyer [?]	$\frac{\alpha a+\beta b+\delta\lvert A\cap B\rvert}{\alpha' a+\beta' b+\delta'\lvert A\cap B\rvert}$ †

† $a = \min\{|A \setminus B|, |B \setminus A|\}$; $b = \max\{|A \setminus B|, |B \setminus A|\}$

with them, the selection of any of these should be made evaluating its performance on the task at hand. The next four coefficients contain parameters that generalize the previous ones and allow the adaptability of the measure to the task. For instance, *generalized Dice* is motivated by the observation that *Dice* and *cosine* coefficients are quotients of $|A \cap B|$ and means between $|A|$ and $|B|$, i.e. arithmetic and geometric mean respectively. Replacing in the denominator of the quotient these means by the generalized mean produce a resemblance coefficient that for different values of p reproduce several existing measures. For instance: for $p = 1$ it is equivalent to Dice, $p \to 0$ to cosine, and $p \to -\infty$ to the overlap coefficient. Similarity, the Tversky's coefficient also generalizes Jaccard's ($\alpha = \beta = 1$) and Dice's ($\alpha = \beta = 0.5$). Jimenez et al. [?] addressed the asymmetry of Tversky's index when $\alpha \neq \beta$ and also re-organize its parameters making α controls the balance between $|A \setminus B|$ and $|B \setminus A|$; and β between $|A \triangle B|$ and $|A \cap B|$. The coefficient proposed by De-Baets and De-Meyer [?] is even more general, but some combinations of the values of its six parameters could yield an expression that does not model similarity. Hence, De-Baets et al. characterized the transitivity of a family of measures for some conditions for the parameters [?].

The parameters of these coefficients can be determined using training data labeled with a gold-standard usually provided by consensus of several human judgments. The common method for obtaining suitable values for parameters is to divide the training data into cross-validation partitions and determine optimal values for the parameters on each partition. Jimenez et al. [?] showed that the performance function for a semantic textual similarity task is mostly convex and smooth w.r.t. the parameters of a resemblance coefficient. Then, the averages of the optimal values for each partition can be used as values for the parameters in test data.

Table 2.4: Different n-grams representations of the word *sequence*

name	**representation**
2-grams	$\{$*se, eq, qu, ue, en, nc, ce*$\}$
3-grams	$\{$*seq, equ, que, uen, enc, nce*$\}$
3-grams padding char.	$\{$***s, *se, seq, equ, que, uen, enc, nce, n***, e**$\}$
skipgrams	$\{$*sq, eu, qe, un, ec, ne*$\}$
2:4-spectra	$\{$*se, eq, qu, ue, en, nc, ce, seq, equ, que, uen, enc, nce, sequ, eque, quen, uenc, ence*$\}$

2.3.2 N-grams representations

N-grams are the collection of consecutive and overlapping sub-strings of length n in a string of length m, with$m > n$. For instance, the collection of 3-grams in the word *sequence* is $\{$*seq, equ, que, uen, enc, nce*$\}$. N-grams represent a partial order of a long string with sub-strings of limited length. The n-grams representation in combination with cardinality-based resemblance coefficients enhances the set-based approach for similarity by incorporating a partial-order notion. The use of n-grams in text applications goes back to the seminal work of Shannon [**?**] to measure the entropy of the English language.

N-grams can be used for lexical similarity by subdividing words into n-grams of characters to cope with OCR errors, misspellings, and inflectional forms either in words as in proper names [**?** **?**]. Several variations of the n-grams representation can be used for lexical comparison (see Table 2.4 for examples). For example, padding characters are additional $n-1$ special characters added to the begin and end of a string [**?**]. These characters are especially useful for distinguishing n-grams that are also common prefixes and suffixes by differentiating them from n-grams in the middle of a character string. Another variation is *skipgrams* that was initially proposed by Pirkola et al.[**?**] defined as non-adjacent 2-grams. At lexical level, skipgrams have shown to outperform n-grams in a cross-lingual spelling variants detection task[**?**].

The selection of n and the type of n-grams representation is usually done by the needs of the task at hand or through training data. For lexical comparison, 2-grams and 3-grams are the most common choice. Jimenez and Gelbukh [**?**] showed that the use of a representation that combines a range of different n-grams (e.g. 2:4-spectra) provides the same or even better performance that the optimal n in the name matching task.

The use of n-grams of words for text comparison is relatively rare in the context of set-based approaches. This is because the use of n-grams of words exponentially increases the size of the vocabulary as n increases. Therefore, this representation is only useful in scenarios involving large corpora, where each n-gram is likely to occur many times.

Table 2.5: Example of set and bag of characters representations

x	set representation	$\|x\|$	bag representation	$\|x\|$
A="*mammal*"	$\{m,a,l\}$	3	$\{m,m,m,a,a,l\}$	6
B="*mall*"	$\{m,a,l\}$	3	$\{m,a,l,l\}$	4
$A \cap B$	$\{m,a,l\}$	3	$\{m,a,l\}$	3
$A \cup B$	$\{m,a,l\}$	3	$\{m,m,m,a,a,l,l\}$	7
$Jaccard(A,B)$	1	-	0.43	-
$Dice(A,B)$	1	-	0.60	-

2.3.3 Bag representations

The set-based representation inherently ignores repetitions of elements in the objects being compared. The theory of bags proposed by Jena and Ghosh [?] is useful for addressing this issue using its definitions for the union and intersection of bags. The intersection of two bags contains the common elements repeated the minimum number of repetitions observed on each bag. Similarity, the union, the maximum frequencies are used. The cardinality property for bags accounts for the total number of elements in the bag including repetitions. Table 2.5 shows an example comparing set and bag of characters representations of the words *mammal* and *mall*. While, using sets both words result identical, bags differentiate these two words.

The bag's definitions of the union, intersection and cardinality can be used analogously with the resemblance coefficients in Table 2.3. Navarro proposed *bag distance* [?], which can be used as a less expensive (in time) approximation of the edit distance (see Subsection 2.6.1). The distance of bags is defined as:

$$DIST_{bag}(A,B) = \max(|A \setminus B|, |B \setminus A|),$$

which can be easily converted into a similarity function by this transformation:

$$SIM_{bag}(A,B) = 1 - \frac{DIST_{bag}(A,B)}{\max(|A|,|B|)}.$$

In our running example (A=*mammal* and B=*mall)*, the bag distance is $DIST_{bag}(A,B) = \max(|mam|,|l|) = 3$, so $SIM_{bag}(A,B) = 0.5$. Note that the edit distance in this example is 4.

2.4 Vector representations

Vector representations consist in representing objects as points (vectors) in a multidimensional space. In this space, objects are compared by geometrical analogies by using distances and angles (Subsection 2.4.1.) This representation is used to represent either words or texts. The common approach to represent words indexes space dimensions by all the possible contexts where any word could occur (i.e. distributional representation [?].) In the case of texts, the approach is known as the vector space model (VSM) [?] where dimensions are indexed by vocabulary words. Unlike set representation, which mainly exploits the information contained in the pair of objects being compared, vector representations usually leverages the information contained in a large collection of objects to provide a comparison between any pair of them. In the case of distributional representations for lexical similarity, this means large corpora where words and contexts occur significantly. For the VSM, this means a large collection of texts. Both large corpora and text collections are used to compile statistics necessary for the determination of the values of the entries in each dimension for each vector. Methods range from simple counts to more sophisticated statistical approaches (Subsections 2.4.3 and 2.4.2.)

Vector representations often involve high dimensionality due to the large number of possible contexts or vocabulary terms. This problem causes sparsity by reducing the chances that two objects being compared share some features (dimensions). This problem can be addressed in two ways: first, the features can be compared with an auxiliary similarity function making it possible to objects that do not have commons features, but have similar ones, to obtain a non-zero similarity score (Subsection 2.4.4); and second, the dimensionality of the space can be reduced exploiting patterns in corpora and text collections using methods such *latent semantic indexing* (LSA) [?] or *non-negative matrix factorization* (NMF) [?]. Recently, a group of methods known as *neural word embedding* [?] aimed to obtain in a single conceptual step low-dimensional representations for words, phrases and short texts.

For illustration of the vectorial representation for lexical and textual similarity, lets consider the following three short texts extracted from the SICK dataset [?]:

- A_1 : "A *young girl* is *dancing*"
- A_2 : "*One young girl* is *standing* on *one leg*"
- A_3 : "A *woman* is *planting* some *flowers*"

In these examples, capitalization and stopwords (shown in small letters) are ignored. To represent these texts as vectors (VSM), the number of dimension and their corresponding labels as extracted from the vocabulary, i.e. {*young, girl, dancing, one, standing, leg, woman, planting, flowers*}. This defines a nine-dimensional space, whose first dimension is indexed by the word *young*, second by *girl*, and so on. Thus, the vector representation for these texts using entries of counts of occurrences on each text is:

$$\begin{aligned} A_1 &= \{1,1,1,0,0,0,0,0,0\} \\ A_2 &= \{1,1,0,2,1,1,0,0,0\} \\ A_3 &= \{0,0,0,0,0,0,1,1,1\} \end{aligned}$$

At this point, bag (see Subsection 2.3.3) and vector -based representations are equivalent, i.e., non-zero entries in a vector model the elements and their number of repetitions in a bag. Likewise, considering only binary occurrences (i.e. if a word occurs once or more in a text, then the vector entry is 1, otherwise is 0) makes set and vector -based representations equivalent.

The matrix $\mathbf{M}_{4\times 9}$, composed by arranging these vectors as matrix rows is known as the *document-term matrix* of a text collection. The transpose of this matrix, $\mathbf{M}^t_{9\times 4}$, provides in its rows a distributional representation for the nine terms $T_1, T_2, \ldots, T_9$ used in the texts (T_1 is *young*, T_2 is *girl* and so on) [**?**]. In this case, each text is considered as a context for the distributional representation of the terms, known as *context-vectors*. Another common type of context is the co-occurrence (or co-location) of two words, i.e., a term T_i occurring next to T_j in a corpus is considered a context for T_j and vice versa [**?**]. So, in our example the distributional representation of the terms is:

$$\begin{array}{rcl} T_1 & = & \{\mathbf{2}, 2, 0, 1, 0, 0, 0, 0, 0\} \\ T_2 & = & \{2, \mathbf{2}, 1, 0, 1, 0, 0, 0, 0\} \\ T_3 & = & \{1, 0, \mathbf{1}, 0, 0, 0, 0, 0, 0\} \\ \vdots & \vdots & \vdots \\ T_9 & = & \{0, 0, 0, 0, 0, 0, 0, 1, \mathbf{1}\} \end{array}$$

The obtained matrix $\mathbf{M}_{9\times 9}$ is known as the *term-context matrix*. In this case, this is a *term-term matrix* whose diagonal contains the number of occurrences of the terms in the corpus.

2.4.1 Similarity and distance

The similarity between two vectors $A = \{a_1, a_2, \ldots, a_k\}$ and $B = \{b_1, b_2, \ldots, b_k\}$ in a k-dimensional space is usually measured by the cosine of the angle between them:

$$SIM_{cosine}(A, B) = \frac{\sum_{i=1}^{k} a_i b_i}{\sqrt{\sum_{i=1}^{k} a_i^2}\sqrt{\sum_{i=1}^{k} b_i^2}}. \tag{2.2}$$

Therefore, two vectors are considered identical (similarity value of 1) if they are parallel, and "completely different" if they are perpendicular (similarity value of 0). Given that in most scenarios entries in the vectors are positive, cosine similarity return values in [0,1] interval. The main property of this measure is that vectors are compared by their direction regardless of its magnitude. In text applications, vector's direction conveys in some way text's meaning and magnitude only its length. In fact, the denominator in Eq 2.2 normalizes the magnitudes of the vectors A and B previous to be compared with the dot product operation.

Another way to compare two vectors in a multidimensional space is measuring the Euclidean distance between them or its generalization the Minkowski distance (Euclidean when $p = 2$):

$$DIST_p(A,B) = \left(\sum_{i=1}^{k}(a_i - b_i)^p\right)^{\frac{1}{p}}$$

However, the property of vector representations that relates direction to "meaning" has made cosine similarity the preferred choice in text applications. Even if the vectors are normalized previous to the computation of their distance, cosine similarity have shown to be better aligned with human judgments of similarity than any other geometrical measure both in distributional and VSM scenarios [**?**].

Similarity and distance (or dissimilarity) are two complementary concepts, so, there are several methods for mapping them. Generally, similarity is defined in a fixed interval (e.g. [0,1] or [-1,1]) and distance is defined as a positive real number. The simplest mapping between these concepts is:

$$SIM(A,B) = \frac{1}{1 + DIST(A,B)}$$

This approach is useful when the distances between objects being compared are not too short nor too large. Another formulation that considers the average pairwise distances between a set of vectors being compared ($\bar{D}$) is:

$$\begin{aligned} SIM(A,B) &= e^{-\alpha \cdot DIST(A,B)} \\ \alpha &= -\frac{(\ln 0.5)}{\bar{D}} \end{aligned}$$

2.4.2 Term weighting approaches for text comparison

One of the most important aspects of the vector representation for text comparison is to decide the values of the entries in vectors. Almost from its inception, VSM has been used with entries more informative than the simple raw occurrences of a term in a document. Earliest approach –still used nowadays– is the *tf-idf* [**?**] approach, which combines the *inverse-document frequency* proposed by Spärk-Jones [**?**] and the *term-frequency* factor. Thus, the entries a_i in A vector are:

$$a_i = tf.idf(w_i, A, \mathrm{D}) = tf(w_i, A).\log\frac{|\mathrm{D}|}{df(w_i, \mathrm{D})} \quad (2.3)$$

Where, w_i is the index term corresponding to the i-th dimension, $tf(w_i, A)$ is the number of occurrences of w_i in A, D is a large document collection and $df(w_i, \mathrm{D})$ is the number of documents in D, in which the term t_i occurs at least once. The motivation for that is to highlight the terms that are frequent into the documents, but rare in the collection. This model was proposed mostly intuitively, but motivated by its effectiveness, over the time various theoretical interpretations has been proposed [**?**]. The field of information retrieval (IR) has contributed with many alternatives for term weighting. One of the most used is the well-known BM25 approach proposed by Robertson et al. [?]:

$$\begin{aligned} a_i &= BM25(w_i, A, \mathrm{D}) = \log\left(\frac{|D| - df(w_i) + 0.5}{df(w_i) + 0.5}\right) \frac{(k_1+1)\times tf(w_i, A)}{k_1\times K(A) + tf(w_i, A)} \\ K(A) &= (1-b) + b\times\frac{len(A)}{avdl(\mathrm{D})} \end{aligned}$$

Where, $k_1 = 1.2$ and $b = 0.75$ are adjustable parameters, $len(A)$ is the number of words in the text represented by A and $avdl(\mathrm{D})$ is the average of the number of words in each document in D. The first factor in BM25 formula is an adjusted *idf* factor; next, k_1 formulates a saturation function that prevents *tf* to grow linearly; and K is a document's length normalization factor controlled by b; when $b = 0$ means no normalization and $b = 1$ full normalization. This model is a practical simplification of a probabilistic model based on a two-Poisson generative model for informative and non-informative words. Although, BM25 is a retrieval formula in the context of documents being retrieved from a collection by a query, these weights also reflect the informativeness of words in the documents where they occur. Hence, this weighting schema has been used widely in the context of vectorial representation for texts [? ?]. Besides, BM25 has preserved its state-of-the-art status for nearly two decades.

2.4.3 Weights for distributional representations

As it was mentioned before in the example at the beginning of this section, a distributional representation for a word may be composed by counting the number of times this word appears in different contexts within a corpus. Although, several aspects such as the size of the corpus and the type of context used affect the quality of distributional representations, the formulations for the weights to be put into vector entries is a fundamental factor. The entries of the vectors that compose the *term-term* matrix depend on the counts of the terms and their co-occurrences. Let t be a target word to be represented and c a context word; N is the number of possible contexts in the corpus; $C(t, c)$ are the number of their co-occurrences and $C(t)$, $C(c)$ are the independent occurrences of t and c. The span of the considered context can be extended by modifying the two-argument function $C(t, c)$ by also counting occurrences of t and c separated until n words. This is equivalent to considering contexts as symmetrical sliding windows of $2n + 1$ size centered on each target word [?]. The total number of contexts N is usually a large number that is close to the total number of words in the corpus. When $C(*)$ and $C(*, *)$ are approximated to the number of hits returned by a search engine on the Web, the value of N is approximated to a very large number (e.g. $N = 10^{10}$ by Bollegala et al. [?]).

One approach to calculate the degree of association between t and c is to treat the counts in the corpus as cardinalities, i.e. $|A| = C(t)$, $|B| = C(c)$ and $|A \cap B| = C(t, c)$. Thus, these cardinalities can be used with any resemblance coefficient [? ?] (see Subsection 2.3.1). However, this approach showed to

Table 2.6: Methods for calculating vector entries for lexical distributional representations using counts.

Approach	**Expression**
Conditional probabilities	$P(c\|t)$
Ratios of conditional probabilities	$\frac{P(c\|t)}{P(c)}$
Point-wise mutual information, PMI	$PMI(c,t) = \log \frac{P(c\|t)}{P(c)}$
Positive PMI [? ?]	$\begin{cases} 0 & \text{if } PMI(c,t) \leq 0 \\ PMI(c,t) & \text{otherwise} \end{cases}$
Normalized PMI [?]	$-\frac{PMI(c,t)}{P(c,t)}$
Local mutual information LMI [? ?]	$C(c,t) \times PMI(c,t)$

be less effective than other probabilistic approaches based on point-wise mutual information (PMI) [?]. For that, the following probabilities can be estimated by maximum likelihood:

$$\begin{aligned} P(c) &= \frac{C(c)}{N} \\ P(t) &= \frac{C(t)}{N} \\ P(c,t) &= \frac{C(t,c)}{N} \\ P(c|t) &= \frac{C(t,c)}{C(t)} \end{aligned}$$

Table 2.6 summarizes the common approaches for calculating the entries of the term-term matrix based on these probabilities. Bullinaria and Levy conducted a comparative study finding that a vectorial representation using positive PMI weights in combination with cosine similarity provides the best results in different lexical semantics tasks. As for the size of the sliding window, they found that $n = 1, 2$ are good choices using weights based on PMI. Once the term-term or term-context matrix is obtained, it is customary to remove or weight dimensions by their deviation, entropy or χ^2 in order to eliminate the effect of highly frequent contexts, and therefore few informative.

2.4.4 Exploiting similarity between dimensions

All the dimensions in any multidimensional space are orthogonal to each other. This may be interpreted as a complete disjuction between them. However, if two different dimensions are indexed by words like *car* and *automobile*, their disjuction is not entirely clear. Less clear is the situation of partially similar or related word pairs like *run/walk* or *doctor/nurse*. This issue can be addressed by using an external source of information that provides a graded notion of similarity between i-th and j-th dimensions, $sim(i,j)$. The idea of the group of approaches discussed in this section is to maintain the structure of the space intact and modify the cosine similarity function, so that exploits $sim(i.j)$ in a meaningful way.

2.4.4.1 SoftTFIDF

The *softTFIDF* measure[?] was motivated by the idea of increasing the cosine similarity value between two texts A and B, represented as vectors with *tf.idf* weights, by adding a weighted contribution of pairs of words $(a,b) : a \in A$, $b \in B$, $a \neq b$ whose lexical similarity is greater than a threshold θ. Moreau et al. [?] noticed that the original definition and implementation provided by Cohen et al. suffered from correctness and normalization problems. However, both original and corrected versions have been widely used primarily to deal with name matching tasks. Thus, the softTFIDF similarity score can be defined as:

$$\begin{aligned}
A &= \{a_1, a_2, \ldots, a_{|A|}\} \\
B &= \{b_1, b_2, \ldots, b_{|B|}\} \\
w(a_i, A) &= \frac{tf.idf(a_i, A)}{\sqrt{\sum_{j=1}^{k} tf.idf(a_j, A)^2}} \\
sim(x,y) &\in [0,1] \\
sim(x,y) &= sim(y,x) \\
sim(x,x) &= 1 \\
sim(x,\emptyset) &= 0 \\
\tilde{b}(\theta, a, B) &= \{b \in B | \forall b' \in B : sim(a,b) \geq sim(a,b') \wedge sim(a,b) > \theta \wedge w(b,B) > 0\} \\
1 &\geq \theta > 0 \\
SIM(A,B) &= \sum_{\{a \in A | w(a,A) > 0\}} w(a,A) \times w(\tilde{b}(\theta,a,B),B) \times sim(a, \tilde{b}(\theta,a,B))
\end{aligned}$$

Where, A and B are two sets of words representing the texts to be compared; $tf.idf(z,Z)$ is the tf.idf weight (Eq. 2.3) of word z in text Z (assuming that Z belongs to a collection of texts); $sim(x,y)$ is the lexical similarity between words x and y; and $closest(\theta,a,B)$ is the most similar element in B to a having $sim(a,b) > \theta$. For name matching, the common choice is $\theta = 0.8$ and the Jaro-Winkler similarity for $sim(x,y)$ (see Section 2.6.3.) The complexity of pair-wise comparisons in a vector space model is $O(\max[|A|,|B|])$, where $|A|$ the number of non-zero entries in A vector. SoftTFIDF increases complexity to $O(|A| \times |B|)$ due to the computation of $\tilde{b}(\theta,a,B)$.

2.4.4.2 Soft cosine

Recently, Sidorov et al. [?] proposed a measure (somehow related to the work of Mikawa et al. [?]) with the same aim as softTFIDF but needless of parameters and free of normalization issues. Given two text represented by A and B in a VSM of dimensionality k and a lexical similarity function $sim(w_i, w_j)$

for comparing the indexing words of dimensions i-th and j-th (s_{ij} for short), the *soft cosine* between them is defined as:

$$SIM(A,B) = \frac{\sum\sum_{i,j}^{k'} s_{ij}a_ib_j}{\sqrt{\sum\sum_{i,j}^{k'} s_{ij}a_ia_j}\sqrt{\sum\sum_{i,j}^{k'} s_{ij}b_ib_j}}$$

Here, a_i and b_j are the entries of the vectors A and B and k' is the dimension of the sub-space where the entries of A and B are not zero in at least one of the vectors. In fact, sub-indexes i and j varies only on such dimensions of the original space. The time complexity of soft cosine is $O(k'^2)$, which is greater that of softTFIDF's but in a similar order. Authors also proposed a method for computing soft cosine in linear time using a transformation matrix that can be obtained offline in $O(k^4)$ time. This approach was originally tested in a question-answering task with multiple choice options, observing that soft cosine consistently outperformed cosine similarity using a $sim(*,*)$ function based on edit distance (see Subsection 2.6.1.)

2.5 Knowledge-based approaches

Knowledge-based approaches exploit a knowledge source external to the data being compared leveraging structured knowledge to provide a notion of similarity according to it. This knowledge source can be a semantic network like WordNet [?] and SNOMED [?] (see Subsection 2.5.1); or encyclopedic resources such as Wikipedia. Knowledge-based approaches have been used to address lexical similarity and textual similarity by combining the scores provided by measures based on knowledge with set or vector -based representations.

2.5.1 Measures based on semantic networks

Semantic networks consist in graphs where nodes correspond to concepts (or words) and edges represent relationships between concepts. These semantic relationships can be either directed (e.g. hypernymy, holonymy) or undirected (e.g. synonymy, antonymy) forming in most cases a tree of a directed-acyclic graph. The intuition for calculating the similarity between pairs of nodes in that graph is that similar concepts should be closer to each other than dissimilar pairs. In addition, the type of modeled relationship on the edges and their directions is also exploited. For instance, most of the existing measures is based on the *is-a* hierarchy formed by hypernym-hyperonym relationships in the graph. Some of the most used measures are described below.

Let a and b be two concepts in a taxonomy where multiple inheritances could happen. The similarity between concepts based on the length of the path that separates them [?] is:

$$sim_{PATH}(a,b) = \frac{1}{path_{LCS}(a,b)}$$

Where, $path_{LCS}(a,b)$ is the minimum number of edges that connects a and b in the graph but visiting their least common subsumer (LCS), which is the common ancestor of a and b that is farther from the top of the is-a taxonomy. This restriction prevents the connection of nodes through multiple inheritance nodes located deeper in the taxonomy. However, McInnes et al. [**?**] relaxed this restriction observing that the measure *u-path* correlated better than sim_{PATH} comparing similarity judgments of physicians using SNOMED[**?**].

Hirst and St. Ongle [**?**] proposed another measure based on $path_{LCS}(a,b)$ in a network connected by not only hypernymy relations but meronymy and antonymy too. Their approach also considers the number of changes of direction, $turns(a,b)$, in the path for hypernyms and meronyms relationships. The measure is given by this expression:

$$sim_{H\&SO}(a,b) = C - path_{LCS}(a,b) - turns(a,b)$$

Where, $C = 8$ and $k = 1$ are constants that can be adjusted to particular tasks and data.

Another group of measures make use of the observation that pairs of concepts separated by a fixed path tend to be less similar than those closer to the top of the taxonomy. Therefore, these measures incorporates information of the total depth of the taxonomy D and the number of edges from a concept c to the top of the taxonomy,$d(c)$. For instance, Leacock and Chodorow [**?**] proposed a measure adjusted by D:

$$sim_{L\&CH}(a,b) = -\log \frac{path_{LCS}(a,b)}{2D}$$

Wu and Palmer [**?**] a measure analogous to the Dice's coefficient but using $d(*)$ as cardinality function:

$$sim_{W\&P}(a,b) = \frac{2 \times d(lcs(a,b))}{d(a) + d(b)}$$

Nguyen and Al-Mubaid [**?**] combined both D and $d(lcs(a,b))$ in the following expression:

$$sim_{N\&AM}(a,b) = \log\left(2 + (path_{LCS}(a,b) - 1) \times (D - d(lcs(a,b)))\right)$$

Choi and Kim incorporated the length of the maximum possible path in the taxonomy, P in a measure that combines paths and depths:

$$sim_{CH\&K}(a,b) = \frac{P - path_{LCH}(a,b)}{P} \times \frac{D - |d(a) - d(b)|}{D}$$

(here, $|*|$ means absolute value).

Batet et al. [**?**] proposed another measure based on the set of "superconcepts" of a concept c, which is the union of the ancestors of the concept c and itself. For the sake of comparison, we present their measure in terms of the components of the measures already presented:

$$sim_{BAT}(a,b) = -\log \frac{d(a)+d(b)-2d(lcs(a,b))}{d(a)+d(b)-d(lcs(a,b))+1}$$

Another important component used for building measures in semantic networks is the *information content* (IC) concept, which was introduced by Resnik [?] for the lexical similarity task. IC came from information theory [?] and is defined as the negative logarithm of the probability of a word or concept, $IC(c) = -\log p(c)$. The probability $p(c)$ can be estimated by maximum likelihood from occurrences of c in a corpus but considering that every occurrence of c should also be counted for any concept up into the is-a taxonomy. Thus, more general concepts become more probable until the top concept in the taxonomy that gets a probability of 1. Given the negative-logarithmic formulation of IC, more general concepts get lower IC and vice versa.

Resnik himself proposed a similarity measure combining IC with LCS [? ?]:

$$sim_{RES}(a,b) = IC(lcs(a,b))$$

Similarly, Lin [?] and Jiang and Conrath proposed a measure combining also the IC for each concept:

$$sim_{LIN}(a,b) = \frac{2IC(lcs(a,b))}{IC(a)+IC(b)}$$

$$sim_{J\&C}(a,b) = \frac{1}{IC(a)+IC(b)-2IC(lcs(a,b))}$$

2.6 String based methods

2.6.1 Levenshtein distance family

Also known as *edit distance* [?], consist in computing the minimal number of edit operations (i.e. *insertion*, *deletion,* and *substitution*) needed to transform a sequence of characters into another. For instance, for transforming *saturday* into *sunday* it requires 3 operations: 0 insertions, 2 deletions, and 1 substitution. Needelman and Wunsch [?] proposed a dynamic programming solution for calculating edit distances efficiently in $O(len(a).len(b))$ time. This algorithm exploits the hierarchical structure of the problem by solving sub-problems splitting each string at all possible positions (see Algorithm 2.1). Although this approach can be applied to long sequences, in text applications it make sense to use it for comparing words, proper names, and short phrases.

The edit distance has been extended by modifying the set of edit operations and adjusting costs associated with each operation. For instance, Damerau [?] proposed *transposition* as an additional edit operation, i.e. swapping of two contiguous elements. This operation allows the modeling of common misspellings

Algorithm 2.1 Needelman and Wunsch's dynamic programming algorithm for edit distance

```
EditDistance(a, b)
    d[0..len(a), 0..len(b)]      //declare matrix of integers
    for i from 0 to len(a)
        d[i, 0] = i
    for j from 1 to len(b)
        d[0, j] = j
    for i from 1 to len(a)
        for j from 1 to len(b)
            if a[i] == b[j] then
                cost = 0
            else
                cost = 1
            d[i, j] = min(d[i-1, j] + 1      // deletion
            d[i, j-1] + 1      // insertion
            d[i-1, j-1] + cost)      // substitution
return (d[m, n])
```

		S	**U**	**N**	**D**	**A**	**Y**
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
A	2	1	1	2	3	3	4
T	3	2	2	2	3	4	4
U	4	3	2	3	3	4	5
R	5	4	3	3	4	4	5
D	6	5	4	4	3	4	5
A	7	6	5	5	4	3	4
Y	8	7	6	6	5	4	3

Figure 2.1: *Edit distance* dynamic programming matrix example.

including missing, erroneous, and transposed characters, comprising most of the mistakes made by humans when writing digital documents. Another variation that obtains the longest common sub-sequence (LCS) [?] can be seen as an edit distance that allows only insertions and deletions at cost 1 (i.e. the number of impaired characters). LCS has been used in text applications not only for comparing character sequences, but also for comparing sequences of words [? ?]. Needleman and Wunsch [?] also proposed a differentiated cost G (i.e. gap cost) for insertion and deletion operations; this approach is also known as the Sellers Algorithm.

Waterman et al. [?] proposed an edit distance measure adding an alphabet distance function $d(c_1, c_2)$, allowing different for editing operations depending on the characters being compared. For instance, the substitution cost for the characters "1" and "*l*" might be lower than the cost between "*e*" and "*h*". Gotoh [?] modified the distance proposed by Waterman et al. considering open/extend gap operations with variable costs (or Smith-Waterman distance [?]). These new operations allow closer matches with truncated or abbreviated strings (e.g. comparing "Michael S. Waterman" with "M. S. Waterman"). In order to have that property, the cost of extending a gap has to be smaller than the gap opening cost.

2.6.2 Learning edit distances

Ristad and Yianilos [?] were the first to propose a method to learn the optimum adaptation cost schema for edit distance for a particular matching problem in a supervised manner. They proposed an edit distance with the three original operations insertion, deletion and substitution. Let a and b be characters in the alphabet Σ of the strings including ϵ as the empty character. The costs to be learned are a matrix $\mathbf{C}_{|\Sigma|\times|\Sigma|}$ whose entries are the substitution costs at character level. The insertion cost for a character a is $c_{a,\epsilon}$. Similarly the deletion and substitution costs are $c_{\epsilon,a}$ and $c_{a,b}$ respectively. The optimal costs are estimated with an expectation maximization (EM) strategy using a training dataset. Bilenko and Mooney [?] proposed a similar approach to learn the optimal cost for an edit distance including open and extend -gap operations.

2.6.3 Jaro and Jaro-Winkler measures

The Jaro's similarity between two strings of length m and n takes only $O(m+n)$ in space and time complexity [?]. It considers the number of common characters c and the number of transpositions t using the following expression.

$$sim_{Jaro}(s_m, s_n) = \frac{1}{3}\left(\frac{c}{m} + \frac{c}{n} + \frac{c-t}{c}\right)$$

Common characters are considered only in a sliding window of size $max(m, n)/2$, in addition, common characters can not be shared and are assigned with a greedy strategy. The case when $c = 0$, sim_{Jaro} must return a value of 0 for avoiding division by zero. In order to compute the transpositions t of two lists, common characters are ordered according with its occurrence in the strings being compared. The number of characters that are unmatched in common positions on both lists is the number of transpositions t.

The Jaro-Winkler similarity [?] is an improved Jaro similarity considering also the number of common characters l at the beginning of both strings using the following expression.

$$sim_{Jaro-Winkler}(s_m, s_n) = sim_{Jaro}(s_m, s_n) + \tfrac{l}{10}\left(1 - sim_{Jaro}(s_m, s_n)\right)$$

Jaro and Jaro-Winkler measures were conceived and have been used mainly for matching proper names. However, these measures can be used as a faster alternative to the edit distance. Similarity, Jaro-Winkler measure have been used for lexical similarity because its property of giving more importance to the mismatches at the characters at the beginning of the words is somehow aligned with the importance of the first characters in the stems of words.

2.7 Other approaches

2.7.1 Monge-Elkan measure

Monge and Elkan [?] proposed a simple but effective method to compare two objects represented as collections of elements. Given two collections of elements A, B and a similarity measure for comparing pairs of elements $sim(a, b)$, the Monge-Elkan measure is computed as follows.

$$SIM_{ME}(A, B) = \frac{1}{|A|}\sum_{i=1}^{|A|} \max{}_{j=1}^{|B|} sim(a_i, b_j) \qquad (2.4)$$

In fact the Monge-Elkan measure is an suboptimal $O(|a| \times |b|)$ approximation to the *assignment problem*, which has a complexity of $O(min(|a|, |b|)^3)$ for the better solutions [?]. Note that Monge-Elkan measure is not symmetrical making of this method rather an inclusion measure than a similarity measure. This feature makes this method useful for detecting directional textual entailment. Nonetheless, it can be symmetrized by $SIM(A, B) = max(SIM_{ME}(A, B), SIM_{ME}(B, A))$.

Jimenez et al. proposed a generalization of the Monge-Elkan method replacing the arithmetic mean in the formula by the generalized mean [?]:

$$SIM_{GME}(A, B) = \left(\frac{1}{|A|}\sum_{i=1}^{|A|}\left(\max{}_{j=1}^{|b|} sim(a_i, b_j)\right)^p\right)^{\frac{1}{p}} \qquad (2.5)$$

Values of $p > 1$ can be interpreted as giving more importance to the pair of elements more similar over the less similar. In a name matching task, a value of $p = 2$ produced optimal results, which can be interpreted as integrating an Euclidean measure as a replacement of the Manhattan distance that was implicitly used in the original formulation. The Monge-Elkan method has been used for comparing names at character level and for short texts using words as elements [? ?].

2.7.2 Compression distance

Cilibrasi and Vitányi [?] proposed the idea of using data compression to obtain a distance measure between two objects A, B using the *normalized compression distance* (NCD) computed with the following expression.

$$NCD(A,B) = \frac{C(A \oplus B) - min\{C(A), C(B)\}}{max\{C(A), C(B)\}}$$

$C(A)$ is the size of the compressed information of A, and $C(A \oplus B)$ is the same function but with the concatenation of A and B. The compressor C has to satisfy the conditions of a *normal* compressor:

1. *Idempotency.* $C(xx) = C(x)$, and $C(\lambda) = 0$, where λ is the empty string.
2. *Monotonicity.* $C(xy) \geq C(x)$.
3. *Symmetry.* $C(xy) = C(yx)$.
4. *Distributivity.* $C(xy) + C(z) \leq C(xz) + C(yz)$.

The data compressor C for text strings and documents may be *BZ2*, *ZLib* or any other data-compression method. This method was originally tested by the authors in document clustering tasks. Christen [?] tested the method in a comparative study of approximate string metrics in name matching tasks obtaining comparable results against the best results obtained with measures of the Jaro's and edit distance families.

Chapter 3

Mathematical properties of Soft Cardinality:

Enhancing Jaccard, Dice and cosine similarity measures with element-wise distance

The soft cardinality function generalizes the concept of counting measure of the classical cardinality of sets. This function provides an intuitive measure of the amount of elements in a collection (i.e. a set or a bag) exploiting the similarities among them. Although soft cardinality was first proposed in an ad-hoc way, it has been successfully used in various tasks in the field of natural language processing (NLP). In this paper, a formal definition of soft cardinality is proposed together with an analysis of the mathematical properties (particularly its boundaries and monotonicity) and an empirical evaluation of the model using synthetic data.

A version of this chapter was submitted to *Information Sciences* journal.

3.1 Introduction

Similarity measures such as Jaccard ($sim(a,b) = {}^{|A\cap B|}/_{|A\cup B|}$) [?], Dice ($sim(a,b) = {}^{2\cdot|A\cap B|}/_{|A|+|B|}$) [?] and cosine ($sim(a,b) = {}^{|A\cap B|}/_{\sqrt{|A|\cdot|B|}}$) [?] coefficients are defined as arithmetic expressions of the cardinality of the sets $|A|$, $|B|$, their union $|A\cup B|$ and intersection $|A\cap B|$. These coefficients are among the most used similarity measures in science. However, this approach fails to capture important characteristics of many data such as the fact there could be a continuous degree of similarity between set's elements. Consider two texts represented as sets that share the same meaning, but have no words in common. Similarity measures based on standard set operations are not able to reflect the similarity between the texts, since these operations fail to represent word similarity. In this paper, this approach is formally extended by making it able to handle this and other scenarios where the similarities between elements induce similarities between collections that contain them.

The soft cardinality model was initially proposed (in 2008) as a convenient way to represent text similarity ([**?**] in Eq. 3.10, and [**?**]), and its effectiveness was validated by the success of the approach to addressing different NLP tasks in SemEval . In this paper, the formal derivation and definition of soft cardinality is presented along with a review of some of its mathematical properties and an experimental validation over synthetic data. This provides a more sound and formal basis for a model that has already been proven effective in practice.

In the spirit of soft cardinality but applied to vectors, Sidorov et al. [**?**] proposed the *soft-cosine* measure, which is particularly similar to our approach. Moreover, Leinster and Cobbold [**?**] (2012) recently proposed a generalized diversity measure in the field of ecology, which incorporates pairwise similarities between species in addition to traditional relative abundance for each species. This measure has the form of the Hill numbers [**?**], which was in turn derived from the generalized Shannon entropy of Renyi [**?**] for different orders q. Our formulation for soft cardinality using default softness parameter ($p = 1$ in Eqs. 3.5 and 3.6) is equivalent to the Leinster-Cobbold diversity of order $q = 0$ (see [**?**]: page 480). Interestingly, both measures cross at that specific point of these parameters, but our derivation came through set theory, while theirs was based on information theory in a direction traced by Ricotta and Szeidl [**?**].

A fundamental property of a collection (set or bag) is its size or cardinality, i.e. the number of elements that compose it. The notion of soft cardinality is to make this count "soft", assuming those elements that are similar to others in the collection "count less" than others that are very different from the rest. Therefore, the soft cardinality of a collection containing similar elements should be less than that of a collection with the same number of elements, but significantly different. For instance, consider two sets of animals: $A = \{$*horse*, *donkey*, *zebra*$\}$ and $B = \{$*horse*, *snake*, *bee*$\}$. The soft cardinality of the former collection could be, say, 1.3 ("a bit more than one animal") and the latter, say, 2.9 ("a bit fewer than three totally different animals"). The classical cardinalities of A and B are identical, but the soft ones provide a more intuitive measure of the "amount" of concepts represented in them.

As shown in the example above, the idea of elements in a collection that count less than others implies that they may contribute with less than 1 to the soft cardinality of the collection. Therefore, the range of this function of cardinality is $\mathbb{R}_+$ instead of $\mathbb{N}$. Furthermore, an element that is identical to another element already in the collection must contribute nothing to the total soft cardinality of the collection. More precisely, all instances (repetitions) of a distinct element contribute only 1 to the soft cardinality of the collection. This idea differs from the previously proposed definitions of the cardinality of a bag, which takes into account all the elements, including duplicates [**?** **?**]. Nevertheless, the number of repetitions of an element in a collection plays an important role in many applications in NLP and IR fields. The soft cardinality model was relaxed to address the above issue by providing a way to handle, besides pairwise similarities, the importance weights associated with each element. Therefore, the number of repetitions may play a role in the calculation of these weights and thus not be ignored by the soft-cardinality model.

The organization of the rest of this paper is as follows: In Section 3.3, the motivation and derivation of the soft cardinality formulation is presented. In Section 3.4, the mathematical definition of soft cardinality is presented. In Section 3.5, a review of the properties is presented. Particularly, we elaborate on the property of monotonicity and the relevance for NLP. Next, in Section 3.7, the behavior and properties of soft cardinality are illustrated with experiments with synthetic data. Finally, in Section 3.8 some brief concluding remarks are provided.

3.2 Intuition

The process that led us to the inception of soft cardinality was purely intuitive. For us, the final model appeared since the first moment. Next, it showed to be useful and competitive in several text processing tasks [**?** **?** **?** **?** **?** **?** **?** **?**]. And finally, the model was analyzed theoretically in an attempt to explain its good performance. The idea came to us intuitively when addressing a name matching task. We observed that in a collection, where repetitions of identical elements are allowed (i.e. a bag), the cardinality of the derived set (i.e. the collection without repetitions) can be expressed by the sum of the inverses of the number of repetitions. For illustration, lets consider the collection $\{a,b,b,c,c,c\}$. The contribution of the element a to the set cardinality was 1, that of each b was $1/2$ and that of each c is $1/3$. Thus, the summation of all contributions $1+1/2+1/2+1/3+1/3+1/3=3$ is the cardinality of the set $\{a,b,c\}$. We arranged the elements in a matrix whose entries indicated if the elements were identical (1) or not (0).

$$\begin{array}{ccccccccc}
 & a & b & b & c & c & c & \textit{row sum} & \textit{inverse} \\
a & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
b & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 1/2 \\
b & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 1/2 \\
c & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 1/3 \\
c & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 1/3 \\
c & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 1/3 \\
 & & & & & & & \sum = & 3
\end{array}$$

The next question was what would happen if the two instances of b were not identical but similar, say 0.9 similar, and the same happens to c. The matrix becomes:

	a	b	b'	c	c'	c''	*row sum*	*inverse*
a	1	0	0	0	0	0	1	1
b	0	1	0.9	0	0	0	1.9	0.53
b'	0	0.9	1	0	0	0	1.9	0.53
c	0	0	0	1	0.9	0.9	2.8	0.36
c'	0	0	0	0.9	1	0.9	2.8	0.36
c''	0	0	0	0.9	0.9	1	2.8	0.36
							$\sum =$	3.12

Does the result 3.12 could mean "three and a bit elements"? The next obvious test was making the instances of b and c almost dissimilar between them, say 0.01:

	a	b	b'	c	c'	c''	*row sum*	*inverse*
a	1	0	0	0	0	0	1	1
b	0	1	0.01	0	0	0	1.01	0.99
b'	0	0.01	1	0	0	0	1.01	0.99
c	0	0	0	1	0.01	0.01	1.02	0.98
c'	0	0	0	0.01	1	0.01	1.02	0.98
c''	0	0	0	0.01	0.01	1	1.02	0.98
							$\sum =$	5.92

Does the result 5.92 could mean "a bit less than six elements"? Our first answer to the previous two questions was yes. Voilá soft cardinality!. This model was so simple and intuitive that our first thought was that most likely the model should already be proposed in the past by someone. The search for connections in the literature was fruitless until the end of 2014 when finally the connection with the work of Leinster and Cobbold was found.

3.3 Derivation

Let $\hat{A} = \{a_1, \ldots, a_n\}$ be a set of sets and $A = \bigcup_{i=1}^{n} a_i$ the set resulting of joining the sets in $\hat{A}$. The cardinality of A is given by the following equation:

$$|A| = \left|\bigcup_{a \in \hat{A}}\right| = \sum_{J \in \mathcal{P}(\hat{A}) \setminus \emptyset} \left(-1^{|J|-1} \left| \bigcap_{a \in J} a \right| \right), \tag{3.1}$$

This is simply the well-known inclusion-exclusion principle. For the case $n = 2$, we can express the cardinality $|A|$ as a function of a the similarity between sets a_1 and a_2 expressed by the Jaccard's coefficient:

$$S(a_1, a_2) = \frac{|a_1 \cap a_2|}{|a_1 \cup a_2|} \tag{3.2}$$

The resulting formula is as follows:

$$\begin{aligned}|A| = |a_1 \cup a_2| &= |a_1| + |a_2| - |a_1 \cap a_2| \\ &= \frac{|a_1|}{S(a_1,a_1)+S(a_1,a_2)} + \frac{|a_2|}{S(a_2,a_2)+S(a_2,a_1)}\end{aligned} \tag{3.3}$$

Each of the summands in Eq. 3.3 represents the individual contributions of a_1 and a_2 to $|A|$. In the event that the sets have zero similarity (their intersection is empty), the individual contributions of them are simply $|a_1|$ and $|a_2|$ (assuming $S(x,x)=1$), and decrease as $S(a_1,a_2)$ increases. This structure is generalized to higher values of n, assuming that the contribution to the total cardinality provided by a set a_i is $|a_i|$ if a_i is not similar to another set but decreases as these similarities increase. The division of $|a_i|$ by the sum of the similarities against all sets in $\hat{A}$ can provide an expression for the contribution of a set following the above intuition. Therefore, the following formula is a plausible approximation for the total cardinality:

$$|A| = \left|\bigcup_{i=1}^{n} a_i\right| \simeq \sum_{i=1}^{n}\left(\frac{|a_i|}{\sum_{j=1}^{n} S(a_i,a_j)}\right). \tag{3.4}$$

Note that Eq. 3.3 and the approximation given in Eq. 3.4 depend solely on pairwise similarities and $|a_i|$.

The exact calculation of the cardinality by Eq. 3.1, requires $\mathcal{O}(2^n)$ time since the cardinality of all the subsets of $\hat{A}$ must be calculated. In contrast, the approximation given by Eq. 3.4 only requires $\mathcal{O}(n^2)$ time. Must importantly, Eq. 3.4 allows for generalization to situations where instead of subsets we have objects that can be compared by a similarity function. As similarities and distances are complementary concepts, that generalization also applies to element-wise distances since any distance function can be transformed into a similarity function.

3.4 Definition

3.4.1 Comparison between elements

Let $A = \{a_{1,}a_2, \ldots, a_n\}$ be a collection whose the elements are comparable between them and repetitions are allowed (also called a bag or multiset). Let us define three notions of comparability among elements, namely *similarity function*, *crisp comparator* and *indistinguishable comparator*.

Definition 1. *A* similarity function *on A is a function $S : A \times A \to [0,1]$ subject to:*

R1. $S(x,x)=1$ any element is identical to itself,

R2. $S(x,y)=1$ iff $x=y$ (x and y are identical),

R3. $S(x, y) = 0$ iff x and y do not share commonalities,

R4. $0 < S(x, y) < 1$ iff x and y are similar to some extent,

R5. $S(x, y) = S(y, x)$ (symmetry).

Definition 2. *A* crisp comparator *for A is a function* $S^\bigstar : A \times A \rightarrow \{0, 1\}$ *subject to R1, R2 and R3, but this last is replaced by* $S(x, y) = 0$ *iff* $x \neq y$ *.*

$$S^\bigstar(x, y) = \begin{cases} 1 & \text{iff } x = y, \\ 0 & \text{iff } x \neq y. \end{cases}$$

Therefore, a crisp comparator is the one that returns a score of 1 only for identical elements and 0 otherwise. The term "crisp" is used here as opposed to "soft", similarly to the use of this term in fuzzy sets, where it means the opposite of "fuzzy" [?].

Definition 3. *An* indistinguishable comparator *in A, is a function* $S^\bullet : A \times A \rightarrow 1$*, i.e.* $\forall x, y : S^\bullet(x, y) = 1$*. An indistinguishable comparator is the one that considers all possible elements identical.*

Proposition 1. Any similarity function S raised to a large power is equivalent to the crisp comparator, i.e. $\lim_{p\rightarrow\infty} S^p = S^\bigstar$. Clearly, if the value returned by S is 1, then the value returned by S^p is 1 for any value of p. In contrast, if the values returned by S satisfies $0 \leq S < 1$, then $\lim_{p\rightarrow\infty} S^p = 0$.

Proposition 2. Any similarity function S raised to a power close to 0 (from the right) is equivalent to the indistinguishable comparator, i.e. $\lim_{p\rightarrow 0^+} S^p = S^\bullet$. Clearly, $\forall S :\ S^0 = 1$ (assuming $0^0 = 1$).

The previous definitions and propositions show that the elements of a collection can be compared with any degree of softness using S^p, obtaining the maximum "softness" (indistinguishability) when $p \rightarrow 0^+$ and the maximum "hardness" (discriminability) when $p \rightarrow \infty$.

3.4.2 Soft cardinality

Definition 4. *Let S be a similarity function to compare pairs of elements, A a collection and a an element in A. The contribution of a to the soft cardinality of A is defined by the formula:*

$$|a|_S = \frac{1}{\sum_{b\in A} S(a, b)^p}; \; p > 0. \tag{3.5}$$

The maximum value of such contribution is 1, obtained when a_i is a fully distinct element in A, i.e. $S(a_{i,}, a_j) = 0$ for any $i \neq j$, and 1 otherwise. Note that by R1 $\forall a_i \in A :\ 0 < |a_i|_S \leq 1$. Here, p becomes a softness control parameter making S to range from $S^\bullet$ to $S^\bigstar$. The default value for this parameter is $p = 1$, which leaves the similarity function S unchanged.

Definition 5. *The soft cardinality of a collection A given a similarity function S, denoted as $|A|_S$, is the addition of all the contributions of their elements:*

$$|A|_S = \sum_{a \in A} |a|_S \,. \tag{3.6}$$

Proposition 3. The soft cardinality of a collection with only one element is 1 and the soft cardinality of the empty collection $\emptyset$ is 0. This is a consequence of Definition 4, Definition 5 and R1.

Corollary 1. The soft cardinality of any collection is always positive, i.e. $\forall A \, \forall S : \, |A|_S \geq 0$.

3.4.3 Disjointness based on similarity

In previous subsections, the term "fully distinct element" was used to refer informally to elements that do not have commonalities against others. In this subsection, similarity-based definitions of disjointness for an element and a sub collection are presented.

Definition 6. *An element a such that $a \in A$ is a disjoint element w.r.t. A if the similarities against all the other elements in A is 0 (excluding a and its repetitions). Complementarily, a is not a disjoint element w.r.t. A if the similarity of a against at least one element in A (again excluding a and its repetitions) is greater than 0.*

Corollary 2. Let a be an element in A. If a is disjoint w.r.t. A, then the contribution $|a|_S$ of a to $|A|_S$ is 1. **Proof.** By Definition 4, $|a|_S$ is the inverse of the addition of the similarities against all the elements in A. If a is disjoint, then all these similarities are 0 except against itself, which is 1. Therefore, the addition of the similarities is 1, so $|a|_S = 1$.

Definition 7. *A sub collection A_d is disjoint in A if all the elements of A_d are disjoint w.r.t. $A \setminus A_d$ and not disjoint w.r.t. A_d. Accordingly, a sub collection containing only one disjoint element w.r.t. A is also a disjoint partition of A.*

Given the above definition, the elements of a disjoint sub collection are linked by similarity relationships, directly or transitively, and disconnected from the other elements in the collection. Therefore, here the concept "disjoint sub collection" is equivalent to the concept "connected component" in graph theory.

Definition 8. *The disjoint partition of a collection A is a set of disjoint sub-collections $D_A = \{A_1, A_2, \ldots, A_{|D_A|}\}$ such that $\bigcup_{A_d \in D_A} A_d = A$ and $\forall i \neq j, A_i \cap A_j = \emptyset$.*

Corollary 3. The soft cardinality of A is additive w.r.t. the disjoint sub collections in D_A. By Definitions 5 and 8 it follows that $|A|_S = \sum_{A_d \in D_A} |A_d|_S$.

3.5 Properties

3.5.1 Bounds

Lemma 1. The soft cardinality is monotonic w.r.t. S. **Proof.** Consider a collection A with more than two elements and two particular elements a_x and a_y that are partially similar, i.e. $0 < S(a_x, a_y) < 1$. The similarity function S can be modified to S_ϵ by altering only the similarity between a_x and a_y by adding a small value ϵ and keeping the other similarities unchanged, i.e. $S_\epsilon(a_x, a_y) = S(a_x, a_y) + \epsilon$ and $S_\epsilon = S$ for the remaining pairs. Let us compare the soft cardinalities $|A|_S$ and $|A|_{S_\epsilon}$. The only element contributions that differ are $|a_x|_{S_\epsilon} = \frac{1}{\frac{1}{|a_x|_S}+\epsilon}$ and $|a_y|_{S_\epsilon} = \frac{1}{\frac{1}{|a_y|_S}+\epsilon}$. Clearly, if $0 < |a|_S \leq 1$ and $\epsilon > 0$, then $|a_x|_{S_\epsilon} < |a_x|_S$ and $|a_y|_{S_\epsilon} < |a_y|_S$, and so, $|A|_{S_\epsilon} < |A|_S$. Conversely, if $\epsilon < 0$, then $|A|_{S_\epsilon} > |A|_S$. Therefore, $|A|_S$ varies monotonically as S varies.

Corollary 4. By Lemma 1, it can be concluded that if there are two similarity functions S_1 and S_2 that satisfy $\forall x, y : S_1(x, y) \geq S_2(x, y)$ (or $S_1 \geq S_2$, for short), then $|A|_{S_1} \leq |A|_{S_2}$. In addition, the opposite, $S_1 \leq S_2 \implies |A|_{S_1} \geq |A|$ is also true. Must be taken into mind that, given the monotonicity of the exponential function, if $p_1 < p_2$, then $S^{p_1} > S^{p_2}$. Therefore, $|A|_{S^{p_1}} < |A|_{S^{p_2}}$, which means that the soft cardinality is monotonic w.r.t. the softness control parameter p.

Both Lemma 1 and Corollary 4 support the intuition that the larger the similarities among the elements of a collection, the lower its soft cardinality. Moreover, as has been demonstrated the monotonicity of the soft cardinality regarding S, and knowing that $S^\star$ and $S^\bullet$ are extremes in the spectrum of possible functions S, boundaries can be found for the soft cardinality.

Lemma 2. The soft cardinality of any disjoint sub collection A_d is lower bounded by 1. **Proof.** The element comparator that provides the maximum degree of similarity is $S^\bullet$. In view of Lemma 1 and Corollary 4, the use of $S^\bullet$ produces the minimum soft cardinality, hence $\forall S : |A_d|_{S^\bullet} \leq |A_d|_S$. Now, by Definition 2 and Definition 4 the contribution of an element $a \in A_d$ is simply $|a|_{S^\bullet} = \frac{1}{|A_d|}$, which leads by Definition 5 to $|A_d|_{S^\bullet} = \sum_{a \in A_d} \frac{1}{|A_d|} = 1$. Therefore, $1 \leq |A_d|_S$.

Theorem 1. The soft cardinality of a collection A is lower bounded by $|A_d| \leq |A|_S$. **Proof.** Suppose that D_A is the disjoint partition of A, by Lemma 3 each disjoint sub collection is lower bounded by $1 \leq |A_d|_S$. This inequality can be added through all disjoint sub collections in A: $\sum_{A_d \in D_A} 1 \leq \sum_{A_d \in D_A} |A_d|_S$. Clearly, the left side of this inequation is $|A_d|$, and by Corollary 3, the right side is $|A|_S$. Therefore the theorem is proved.

Lemma 3. The soft cardinality of any disjoint sub collection A_d is upper bounded by $|A_d|$. **Proof.** Analogously as in Lemma 2 , the element comparator that provides the minimum degree of similarity is $S^\star$, which produces the maximum soft cardinality. Besides, by Definition 4, $|a|_{S^\star} = \frac{1}{\sum_{b \in A_d} S^\star(a,b)^p} = 1$

because $S^{\star}(a,b)$ is only 1 if $a=b$ and 0 otherwise. Thus, by Definition 5, $|A_d|_{S^\star} = \sum_{a\in A_d} |a|_{S^\star} = |A_d|$. Finally, by Corollary 4, if $\forall x,y : S(x,y) \geq S^{\star}(x,y)$, then $|A_d|_S \leq |A_d|_{S^\star}$. Therefore $|A_d|_S \leq |A_d|$.

Theorem 2. The soft cardinality of a collection A is upper bounded by $|A|_S \leq |A|$. **Proof.** Suppose that D_A is the disjoint partition of A, by Lemma 3 each disjoint sub collection is bounded by $|A_d|_S \leq |A_d|$. Similarly as in Theorem 1, this inequality can be added through all disjoint sub collections in A: $\sum_{A_d\in D_A} |A_d|_S \leq \sum_{A_d\in D_A} |A_d|$. The left side of the inequality is $|A|_S$ due to Corollary 3. The right side is $|A|$ due to Definition 8, which prevents any element to be counted more than once. Therefore, $|A|_S \leq |A|$.

In conclusion for any collection A and its disjoint partition D_A, the soft cardinality is bounded by $|D_A| \leq |A|_S \leq |A|$. Furthermore, it is interesting to see that the classical view of a set is equivalent to a collection that has only disjoint elements. Accordingly, for such collection $|A| = |A|_{S^\star}$, showing that the soft cardinality is a generalization of the classical cardinality. Besides, it was shown that in the case when soft cardinality is equivalent to classic cardinality the latter is an upper bound for the former. This result agrees with our initial intuition that if there are commonalities among the elements of A, then the soft cardinality must be less than the simple counting of the elements. Similarly, since the lower bound is the number of disjoint sub collections, it is shown that the soft cardinality can decrease only until a meaningful limit. Thus, if a collection has only one disjoint sub-collection, then the minimal soft cardinality is 1, i.e. one element.

3.5.2 Monotonicity

Up to here, the term monotonicity was used in the sense of the property of a function w.r.t. its parameters, i.e. $|A|_S$ with respect to S or p. However, monotonicity is a mathematical property of a measure at a playground, where a numeric value assigned by the measure to any subset must be less than or equal to the numeric value assigned to the set. For instance, the classical cardinality is a trivial counting measure that fulfills monotonicity. Thus, the measure that the soft cardinality tries to approximate (Eq. 3.4) is also monotonic. Intuitively, the soft cardinality should satisfy monotonicity because it has been shown that it is a softened enumeration of elements in a collection. It makes sense that, when an element is added to a collection, the soft cardinality must increase or remain unchanged, but never decrease. However, it was observed empirically that the soft cardinality observes monotonicity in most cases, but “fails” in some situations. In fact, in Section 3.7 , it can be seen that the number of cases where the soft cardinality is not monotonic decrease as the softness control parameter p increases. In this sub section, the cases in which the soft cardinality is monotonic or not, are reviewed analytically. In particular, a non-monotonic example may show that this behavior models heuristics that could be an advantage for NLP applications rather than a drawback.

The following statements require these assumptions: Let A be a collection and b a new element such that $b \notin A$. The insertion of b into A results in a new collection, denoted by $A \oplus b$.

Theorem 3. If b is similar to only one element a such that $a \in A$, then $|A \oplus b|_S \geq |A|_S$. **Proof.** The contributions to the soft cardinality of the common elements between A and $A \oplus b$ are identical except for a, that is $|a|_S$ in A, and decrease to $\frac{|a|_S}{1+|a|_S S(a,b)}$ in $A \oplus b$. Besides, $|A \oplus b|_S$ increases in comparison with $|A|_S$ because the contribution of the new element b, which is $|b|_S = \frac{1}{1+S(a,b)}$. Hence, $|A \oplus b|_S \geq |A|_S$ iff $\frac{1}{1+S(a,b)} \geq |a|_S - \frac{|a|_S}{1+|a|_S S(a,b)}$, i.e. iff the increment at the left side is larger than the excrement at right. This inequation can be rewritten as $\frac{1}{|a|_S(1+S(a,b))} + \frac{1}{1+|a|_S S(a,b)} \geq 1$ (recall that $0 < |a|_S \leq 1$ and $0 < S(a,b) \leq 1$). In the previous inequation, the first addend gets the minimal value of 0.5 if $|a|_S = S(a,b) = 1$. Similarly, the second addend gets the same minimal value of 0.5 also if $S(a,b) = 1$, for a total at the left side of the inequation of 1. Therefore, the inequality holds for all possible values of $|a|_S$ and $S(a,b)$. Consequently, the theorem is proven.

Remark 1. *If b is disjoint w.r.t. $A \oplus b$, then $|A \oplus b|_S = |A|_S + 1$. This property is equivalent to the classical cardinality because $|b| = 1$ and all the contributions of the elements of A to the soft cardinality of the collection are unchanged.*

Remark 2. *If $a \in A$, a is disjoint w.r.t. A, and $S(a,b) = 1$, then $|A \oplus b|_S = |A|_S$. Exact copies of disjoint elements do not increase the soft cardinality.*

Theorem 3 and Remarks 1 and 2 cover the cases when the soft cardinality is strictly monotonic, being Remark 1 the case with maximal monotonicity and Remark 2 the minimal.

Theorem 4. If b is similar to more than one element in A, then $|A \oplus b|_S$ could be less, equal or greater than $|A|_S$. **Proof.** Let C be a sub collection of A (i.e. $C \subseteq A$) such that b is similar to all the elements in C (i.e. $\forall a \in C : S(a,b) > 0$), and let m be the number of such elements (i.e. $m = |C| \geq 2$). Similar as in Theorem 3, $|A \oplus b|_S$ differs from $|A|_S$ by an increment due to $|b|_S$, and also by the decrements in the contributions of the elements of C due to their similarities against b. In the following questioned equality, the increment is put at left and the absolute decrements at right:

$$\frac{1}{1 + \sum_{a \in C} S(a,b)} \overset{?}{=} \sum_{a \in C} \left[|a|_S - \frac{|a|_S}{1 + |a|_S S(a,b)} \right]. \tag{3.7}$$

Here, if the left side is equal or greater than the right side, then the insertion of b holds monotonicity. Otherwise, the insertion of b into A produces a reduction in the soft cardinality. To prove this theorem, the following three examples illustrate the feasibility of each case of the questioned equality in Eq. 3.7:

Example 1, for $|A \oplus b|_S < |A|_S$. Suppose that $\forall a \in C : |a|_S = 1$ and $S(a,b) = 1$. The replacement of these values in Eq. 3.7 produces: $\frac{1}{1+m} \overset{?}{=} \frac{m}{2}$. As

$m \geq 2$, it follows that $\frac{1}{1+m} < \frac{m}{2}$ for $m \geq 2$, so $|A \oplus b|_S < |A|_S$ (not monotonic case).

Example 2, for $|A \oplus b|_S = |A|_S$. Suppose that $\forall a \in C : |a|_S = \frac{1}{m}$and $S(a,b) = 1$. The replacement of these values in Eq. 3.7 produces: $\frac{1}{1+m} \overset{?}{=} \frac{1}{1+m}$. Obviously, $|A_{b^+}|_S = |A|_S$. Interestingly, if the similarity is set to a fixed value $S(a,b) = k$, then it gets $\frac{1}{1+mk} \geq \frac{k}{k+m}$, which is true for $m \geq 2$ and $0 \leq k \leq 1$. In this case, $|A \oplus b|_S \geq |A|_S$ (monotonic case).

Example 3, for $|A \oplus b|_S > |A|_S$. Suppose that $\forall a \in C : |a|_S = \frac{1}{1+m}$ and $S(a,b) = 1$. The replacement of these values in Eq. 3.7 produces: $\frac{1}{1+m} \overset{?}{=} \frac{m}{1+m} - \frac{m}{2+m}$. The questioned equality can be rewritten as $\frac{1-m}{1+m} \overset{?}{=} \frac{-m}{2+m}$. Let $m' = m - 1$, when m is replaced by m' in the right side it gets $\frac{1-m}{1+m} \overset{?}{=} \frac{1-m'}{1+m'}$. Now, it is easier to see that on both sides produce the same series of numbers but with an offset of 1. Consequently, $\frac{1-m}{1+m} > \frac{-m}{2+m}$ for $m \geq 2$, and $|A \oplus b|_S > |A|_S$ (monotonic case).

Note that if a new inserted element has similarities with the elements belonging to different disjoint sub collections, these sub collections become one disjoint sub collection. Being reduced the number of disjoint sub collections, the lower boundary of the soft cardinality is also reduced (by Theorem 1) thus compromising monotonicity.

Theorem 4 shows an interesting fact: when the new element being added to the collection is similar to more than one element in the collection, the monotonicity of the soft cardinality may or may not hold. Contrary, as shown in Theorem 3 , when the new element is similar to only one element in the collection, monotonicity is strict. Furthermore, the above examples in Theorem 4 show that monotonicity depends on these multiple similarities as on the contributions to the soft cardinality of the elements similar to the new one. Apparently, when these contributions and/or these similarities are low, monotonicity seems to hold. Unlike this, when the new element has similarities close to 1 against other elements, and these elements are almost disjoint (the contribution to the soft cardinality $|a|_S$ close to 1), the situation leads to non-monotonicity. This scenario rise a question: Is this behavior useful or harmful for the natural language processing?

To address the this question, let us first say that an element a_1 in A with a high contribution to the soft cardinality (i.e. $|a_1|_S$ is close to 1) means that such element is mostly not similar against the others elements in A. If another element a_2 in A has the same condition, then it is probable that the similarity between a_1 and a_2 is also low. In that scenario, if an element b is being added to A, and it happens that $S(a_1, b)$ and $S(a_2, b)$ are both high (close to 1), then b provides new evidence that a_1 and a_2 are similar. Clearly, this situation does not fulfill the transitive closure of S because a_1 is similar to b, b to a_2, but a_1 and a_2 are not similar. Although, S is not restricted with metric properties, the triangle inequality in the dissimilarities $(1-S)$ may be violated too. Contrarily, if the element being added is only similar to one element in the collection, no new similarity connections are made inside that collection, and so monotonicity

Table 3.1: Relatedness scores from Google Books Ngram corpus (1999-2000) using nPMI in a sliding window of three words

		$S(a,b)$			
$a\downarrow$	$b\rightarrow$	*New*	*York*	*City*	*Times*
New		1.0000	0.9128	0.5207	0.5920
York		0.9128	1.0000	0.5542	0.6283
City		0.5207	0.5542	1.0000	0.1508
Times		0.5920	0.6283	0.1508	1.0000

Table 3.2: Soft cardinality scores for all sub collections of {*New*, *York*, *City*, *Times*}

i	A_i	$\|A_i\|_S$	i	A_i	$\|A_i\|_S$
0	{}	0.0000	8	{*York*, *City*}	1.2868
1	{*New*}	1.0000	9	{*York*, *Times*}	1.2283
2	{*York*}	1.0000	10	{*New*, *York*}	1.0456
3	{*City*}	1.0000	11	{*New*, *York*, *City*}	1.2983
4	{*Times*}	1.0000	12	{*New*, *York*, *Times*}	1.2432
5	{*City*, *Times*}	**1.7380**	13	{*New*, *City*, *Times*}	1.6454
6	{*New*, *City*}	1.3152	14	{*York*, *City*, *Times*}	1.6068
7	{*New*, *Times*}	1.2563	15	{*York*, *New*, *City*, *Times*}	1.5247

holds as shown by Theorem 3.

Now, let us illustrate the pseudo-monotonic behavior of the soft cardinality using an example. Consider the collection of words A={*New*, *York*, *City*, *Times*} and a similarity function S obtained by the normalized pointwise mutual information (nPMI) [**?**] using counts from the Google Books Ngram corpus[1] between years 1999 and 2000, an sliding windows of size 3, and $p = 1$. The resulting collocational-relatedness function is shown in Table 3.1.

Although, only collocational evidence is used here, the S function seems to follow intuition. In such small context, terms like *City* and *Times* are almost unrelated, which is reasonable if no further context is provided. Once, the term *New* is added to the collection, it provides new evidence of the second-order relatedness between *City* and *Times*. Similarly, the term *York* strengthen this evidence. Table 3.2 shows the soft cardinality scores obtained by Eq. 3.6 for all sub collections in A. Note that in this example, as S is a relatedness function, the soft cardinality measures the amount of unrelated terms.

Among these scores, sub collections such as A_{13} and A_{14}, which have more words than A_5, get a lower soft cardinality evidencing a non-monotonic behavior. Besides, A_{15} gets an even lower soft cardinality showing how the additional context reinforces the relationship between *City* and *Times*. Similarly, A_6 to A_{11} and A_7 to A_{12} are non-monotonic insertions. It is interesting to see that, in this example all the other possible element insertions (e.g. $|A_{11}|_S > |A_8|_S$) the

[1]https://books.google.com/ngrams

monotonicity holds, showing that the soft cardinality can deal with transitive and non-transitive relationships in A induced by S. Clearly, for this particular example a non-strictly monotonic measure is more convenient than a monotonic one. Besides, this example illustrates how all the n-wise relationships in a collection can be inferred from pairwise ones using the soft cardinality.

The previous example also suggests, that the amount of information in natural language is not a measure (in mathematical sense) with respect to the number of words. It is generally accepted that more words in a message not necessarily convey more concepts or information. For instance, words and its definitions exhibit a non-monotonic behavior. Consider the text "a man who works" conveys at least two distinct concepts, namely: "man" and "to work". However, if the word "wood" is added to the message the definition becomes collapsed in only one concept, "carpenter".

3.5.3 Idempotence

Theorem 5. Let A and B be two collections that contain the same number of distinct elements but differ in the total number the elements (counting repetitions). If the relative proportions of the number of repetitions of the elements in A and B are the same, then $|A|_S = |B|_S$. **Proof.** Let A be a collection with m distinct elements $a_1, a_2, \dots, a_m$ (i.e. $\forall i, j \in \{1, 2, \dots, m\} \wedge i \neq j, S(a_i, a_j) < 1$) and r_i is the number of repetitions of a_i in A. The soft cardinality of A becomes:

$$|A|_S = \sum_{i=1}^{m} \frac{r_i}{\sum_{j=1}^{m} r_j \cdot S(a_i, a_j)^p}.$$

The number of elements in A is $n = \sum_i^m r_i$ and the relative proportion of a_i is $f_i = {r_i}/{n}$. Replacing r_i by $n \cdot f_i$ in the previous equation gets:

$$|A|_S = \sum_{i=1}^{m} \frac{f_i}{\sum_{j=1}^{m} f_j \cdot S(a_i, a_j)^p}.$$

Therefore, if the relative proportions f_i are the same in A and B, then $|A|_S = |B|_S$.

Theorem 5 shows that soft cardinality depends (besides on similarities among elements) either on the relative proportions of the repetitions of the elements or the number of repetitions. Therefore, if B contains the same elements of A but repeated k times, then the soft cardinality of both collections is the same. Working with relative proportions could be more convenient in applications such as document and image processing making the results independent of the length or resolution respectively.

3.6 Similarity functions

3.6.1 Inferring cardinality of intersection

The soft cardinality of the intersection of two collections cannot be calculated directly from $A \cap B$ because the intersection operator is inherently crisp. This means that, if there are no common elements between A and B, their intersection is empty, and so its soft cardinality is 0. The following definition allows to infer the soft cardinality of the intersection between two collections by means of the soft cardinalities of each collection and their union.

Definition 9. *Let A and B be two collections, the soft cardinality of the intersection of them is $|A \cap B|_S = |A|_S + |B|_S - |A \cup B|_S$. In this case, the operator $\cup$ means* bag union*, which takes the maximum of the number of occurrences of the common elements in each bag. Example: $\{1, 1, 2, 3\} \cup \{1, 2, 2\} = \{1, 1, 2, 2, 3\}$.*

This allows non-empty intersections for pairs of collections that have no elements in common, but they do have similar ones. Once $|A \cup B|_S$, $|A \cap B|_S$, $|A|_S$ and $|B|_S$ are known, it is possible to obtain all the other areas in the Venn's diagram of two sets, i.e. $|A \triangle B|_S = |A \cup B|_S - |A \cap B|_S$, $|A \setminus B|_S = |A|_S - |A \cap B|_S$ and $|B \setminus A|_S = |B|_S - |A \cap B|_S$. These are the building blocks of almost all cardinality-based resemblance coefficients.

3.6.2 Building similarity functions

There are several parametric and non-parametric families of cardinality based similarity functions ranging from simple formulations such as Jaccard's [?] and Dice's [?] coefficients until parametric formulations [? ?]. The way to build a text similarity function is *i)* to select a linguistic unit to be compared (e.g. sentences), *ii)* to use a representation of the texts based on bags (e.g. bags of words, n-grams, dependencies, etc.), *iii)* to choose a cardinality based similarity coefficient (e.g. Jaccard's, Tversky's, De Baet's coefficients), and *iv)* to provide a pairwise similarity function S for the elements produced by the used text representation (e.g. normalized Levenshtein similarity, nPMI [?], normalized path length in WordNet [?], etc.). The simplest example of such similarity function for pairs of sentences is:

$$S_{sentence}(A, B) = \frac{|A \cap B|_{S_{word}}}{|A \cup B|_{S_{word}}}. \tag{3.8}$$

Here, $S_{sentence}$ is the produced similarity function that compares pairs of text sentences based on the similarities between words provided by S_{word}. The only parameter to be adjusted in Eq. 3.8 is p. Jimenez et al. [?] showed that the default $p = 1$ works fine for short sentences in English. However, a convenient value for p depends mainly on the range and distribution of the values returned by S_{word}, on the length of the texts, and on the task at hand. An optimal value can be determined using training data with a gold standard of similarity scores. Other parameters introduced by the use of parametric cardinality-based coefficients, such as α and β in the Tversky's index [?], can be determined in the same way [?].

It is important to note that Eq. 3.8 is recursive, similar to the popular Monge-Elkan measure [? ?]. That is, the similarity function $S_{sentence}$ is obtained from another similarity function, S_{word}. This idea can be recursively used to build a similarity function $S_{paragraph}$ from $S_{sentence}$, and so on. In that way, it is possible to build similarity functions that exploit the hierarchical structure of text and natural language.

The previous example the ilustrates the process for contructing a similarity function using soft cardinality. Clearly, the process is analogous in any other domain where objects can be modeled as collections of elements.

3.6.3 Weighted soft cardinality

Now, let us introduce a relaxed soft-cardinality model that allows importance weights associated to the elements of a collection.

Definition 10. *The weighted contribution of an element a, such that $a \in A$, to $|A|_S$, is given by multiplying the unweighted contribution by a positive weighting function having $\forall x \in A : W(x) \geq 0$, that is:*

$$|a|_S = \frac{W(a)}{\sum_{x \in A \wedge W(x)>0} S(a,x)}. \tag{3.9}$$

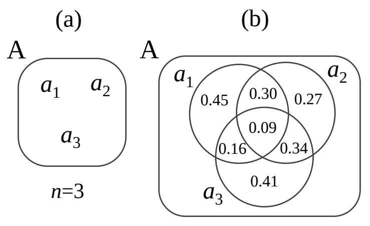
The function $W(a_i)$ is any weighting mechanism for the elements of A. When the elements of A are divisible into sub elements, the straightforward choice is $W(x) = |x|$. Another examples could be entropy $W(x) = -P(x).\log_2(P(x))$ or the *tf-idf* weighting schema $W(x, A) = tf(x, A).idf(x, U)$ [?], here U is a large set of collections or words, i.e. a document collection.

This weighted version of the soft cardinality allows incorporating the common practice in NLP of promoting informative words and ignore non-informative words. It is important to note, that elements with null weights (or close to 0) should be removed from the collections because, even though their contribution is 0, their similarities still interacts with the other elements affecting their contributions. This issue reveals the fact that most of the properties of the soft cardinality get overwritten because of the effect of the weights. Despite that, the weighted approach preserves the original motivations of the soft cardinality and extends its modeling ability.

3.7 Testing soft cardinality in random data

In this section, the properties of the soft cardinality are empirically explored using random data.

The first experiment aims to compare the soft cardinality against a gold standard that provides exact soft cardinality. For a given number n of elements in a collection, all their m-wise ($m \in \{2, 3, \ldots, n\}$) relations can be randomly



(c)

$|a_1|$=0.45+0.30+0.09+0.16 = 1.00

$|a_2|$=0.27+0.34+0.09+0.30 = 1.00

$|a_3|$=0.41+0.16+0.09+0.34 = 1.00

$|A|_*$=0.45+0.30+0.27+0.34+
0.41+0.16+0.9 = 2.02

(d)

$S(x,y)=|x \cap y| \,/\, (|x|+|y|-|x \cap y|)$

$|a_1 \cap a_2|$=0.30+0.09=0.39

$|a_2 \cap a_3|$=0.34+0.09=0.43 ;

$|a_1 \cap a_3|$=0.16+0.09=0.25

(e)

S	a_1	a_2	a_3	$1/\|a_i\|_S$	$\|a_i\|_S$
a_1	1.0	0.242	0.143	1.385	0.722
a_2	0.242	1.0	0.274	1.516	0.660
a_3	0.143	0.274	1.0	1.417	0.706

$|A|_S$=0.722+0.660+0.706= 2.088 $|A|_* \approx |A|_S$

Figure 3.1: Example of obtaining a gold standard for comparison with the soft cardinality

generated assigning to them uniformly distributed random numbers drawn from the [0,1] interval. These m-wise relations can be seen as all distinct areas of a Venn's diagram built using the n elements treated as sets. The random numbers are adjusted in order to make the individual areas that comprise a_1, a_2 and a_3 add up to 1. Thus, the summation of all those relations values provides a measure of the amount of non-redundant information in the collection. Figure 3.1 (a) to (c) depicts an example of this process for $n = 3$: (a) a collection with 3 elements is given, (b) each element is treated as a set and a random number is assigned to each possible m-wise relation, (c) the "cardinality" of each element is checked to add up to 1. In that example, it is possible to say that the soft counting of the number of elements in that collection is 2.02 elements, which is analogous to the idea of soft cardinality, but computed using all possible relations among the elements. Note that the number of m-wise relations is $2^n - 1$, which grows exponentially making the generation of this gold standard only feasible for relatively small values of n (until $n = 20$ in our experiments). For instance, for $n = 20$, while soft cardinality requires only 190 pairwise relationships, the gold standard requires 1,048,576 m-wise.

Once the gold standard value ($|A|_*$ in Figure 3.1(c)) is obtained for the collection, it is compared against the soft cardinality. For that purpose, the pairwise intersections, which are only ${}^{n(n-1)}\!/_2$, are extracted from the same random data and the Jaccard coefficient is used to get the element-to-element similarity function S. Finally, the soft cardinality is computed (Eq. 3.6 using $p = 1$) and compared against the gold standard. This process is shown for our running example in Figure 3.1 (d) and (e). In this example, the gold standard and soft cardinality are relatively close ($2.088 \approx 2.02$).

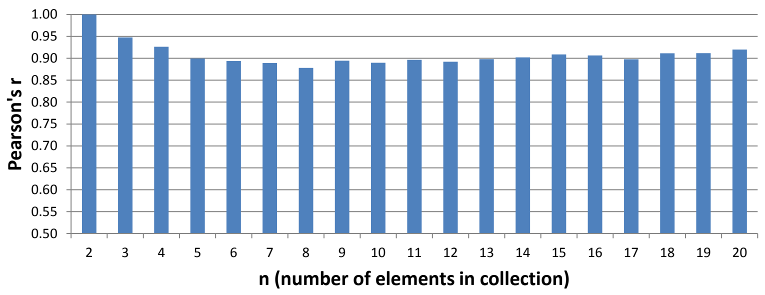


Figure 3.2: Correlations obtained between the soft cardinality and the gold standard for 1,000 samples varying n from 2 to 20

This process is repeated 1,000 times generating different random data, and all obtained values for the gold standard and soft cardinality are compared using the Pearson's correlation. Figure 3.2 shows the resulting correlations for collections ranging from $n = 2$ to $n = 20$. In addition, Figure 3.3 shows plots of the data for 6 selected values of n.

These results show that the soft cardinality is highly correlated with the proposed gold standard. At first, the correlation seems to decrease as n increases, but then it stabilizes around 0.9, which is highly significant for a sample size of 1,000. If the soft cardinality is interpreted as a pairwise approximation of the "real" one computed with all m-wise relationships, then this stable correlation suggest that the soft cardinality is a good approximation for those values.

The results of this experiment could depend importantly on how the random data was drawn. Clearly, using uniformly distributed random numbers for each distinct area of the Venn's diagram (see Figure [fig:example-exact-sc](b)) promotes high degrees of similarity among the elements. As it can be seen in Figure [fig:correlation-plots], as n increases the data varies in smaller ranges, which means that the degree of redundancy is higher in larger collections using that uniformly generated random data. Nevertheless, it is noticeable that the correlation between the soft cardinality and the proposed gold standard is preserved even at such small scale. Additional experiments were carried out with other distributions observing that such correlation prevails. For instance, in Figure [fig:non_uniform_random_plots] the uniformly generated data (middle points in blue) is compared with one set generated by promoting similarities (lower points in red) and other promoting dissimilarities (upper points in green). The dissimilarities were promoted by dividing each random number r by a factor that depends on the number q of elements involved in the relation, by the expression: $r' = {}^{r}/_{(1+10q)}$. Similarly, for promoting similarities the expression was $r' = {}^{r}/_{(1+10(n-q))}$.

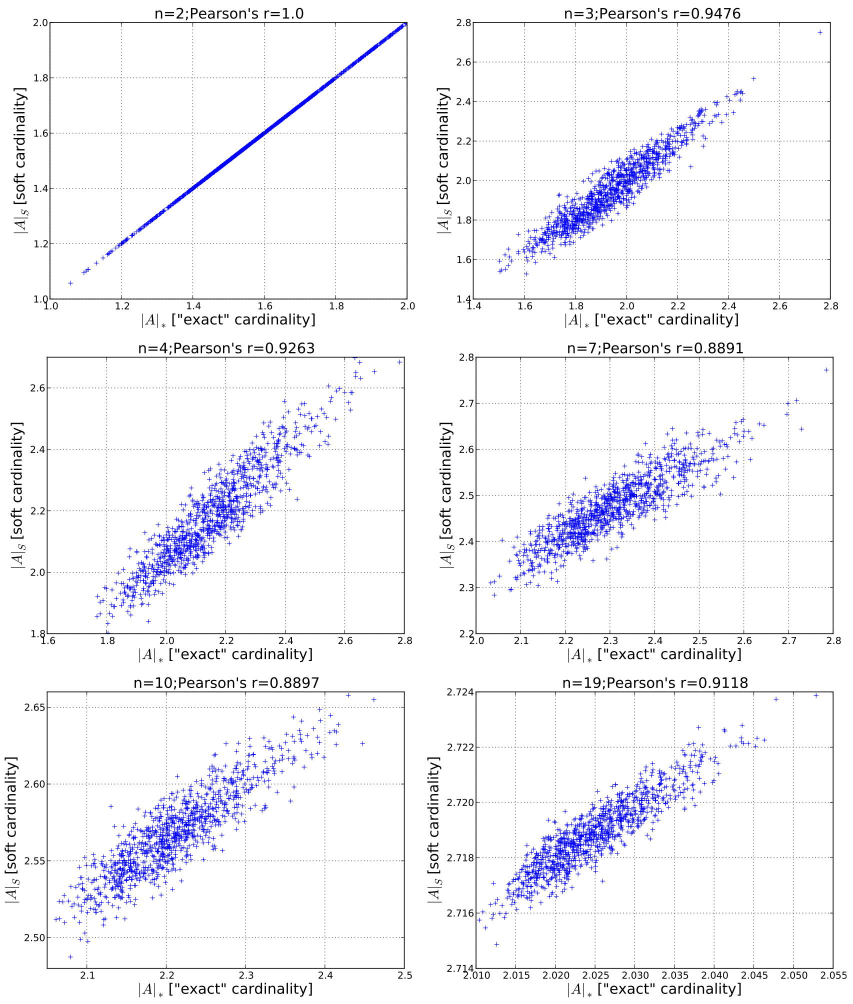


Figure 3.3: Correlation between "exact" and soft cardinality for 1,000 random collections of n elements

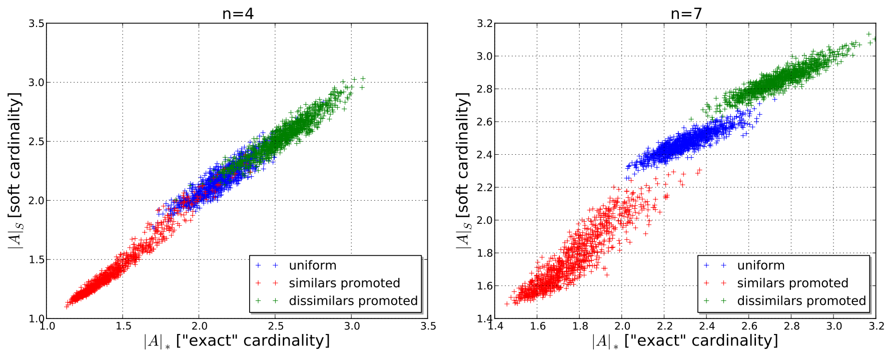


Figure 3.4: Correlation between “exact” and soft cardinalities for differently generated random data

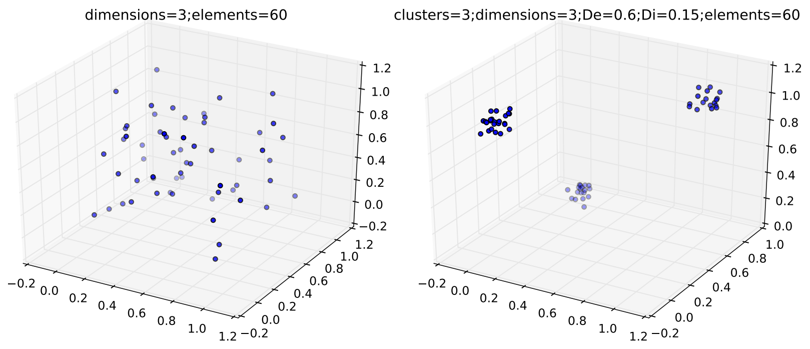


Figure 3.5: Example of 60 random uniform points at left, and clustered in 3 groups at right

3.7.1 Pseudo-monotonicity and softness control

The goal of this subsection is to visualize the pseudo-monotonic behavior of the soft cardinality w.r.t. the softness control parameter p in randomly generated data. For that, several sets of 60 points are generated in a tridimensional space inside a cube with a diagonal from (0,0,0) to (1,1,1). For each run, two sets are generated: one uniformly and another clustered in three zones. The clustered data was generated by, first selecting three random centroids separated at least by a distance D_e, and then iteratively generating random points around them into spheres of radius D_i. Examples of these data sets are shown in Figure 3.5, uniform at left and clustered at right.

The similarity function for pairs of points is based on the Euclidean distance:

$$S_{points}(a,b) = 1 - \frac{distance(a,b)}{\sqrt{3}}.$$

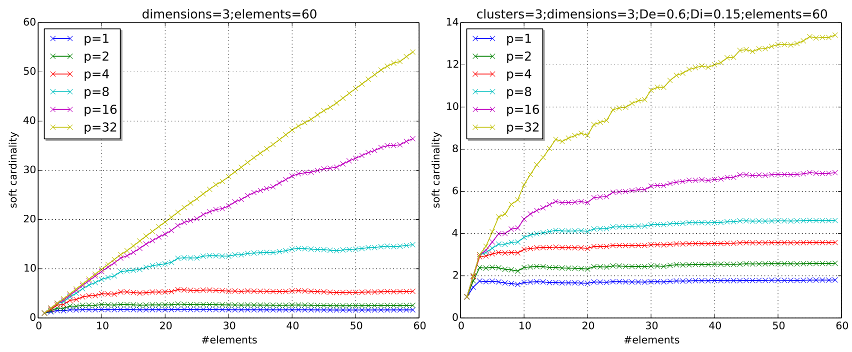


Figure 3.6: Soft cardinalities of subsets of incremental number of random generated elements; uniformly at left and 3-clustered at right

Using this similarity function, the soft cardinalities of sets with incremental insertions of elements are calculated using different values of the softness control parameter p. In Figure 3.6, each series shows the results for a different run. This figure shows that most of the element insertions, both in the uniform and clustered cases, produce monotonic increments in the soft cardinality. Although, there are some non-monotonic insertions, the average tendency of each series is clearly monotonic. It is interesting to see that the values of the soft cardinality in the clustered data are considerably lower than those of the uniform data, considering that the average similarity in both data sets is similar. This shows that the soft cardinality reveals that the "amount of information" in a set is larger when it is generated with a criterion of maximum entropy (uniform). Figure 3.6 (at left) also shows how for larger values of p the soft cardinality tends to its upper boundary, i.e. the classical cardinality.

The upper and lower boundaries of the soft cardinality can be better appreciated in Figure 3.7 at left, where the softness control parameter p is plotted versus the soft cardinality of different sized sets. In the uniform data, the lower boundary is obtained with the minimum value of p. Next, as p increases each series tends asymptotically to the upper boundary, i.e. the number of elements. It is interesting to see that, in the clustered data (Figure 3.7 at right) there are local minima in the slopes of the curves when the soft cardinality is around 3, i.e. the number of clusters. These plots also confirm that the soft cardinality is strictly monotonic with respect to the softness control parameter (Lemma 1).

The results discussed in this subsection confirmed empirically and graphically some of the properties of the soft cardinality properties theoretically presented earlier.

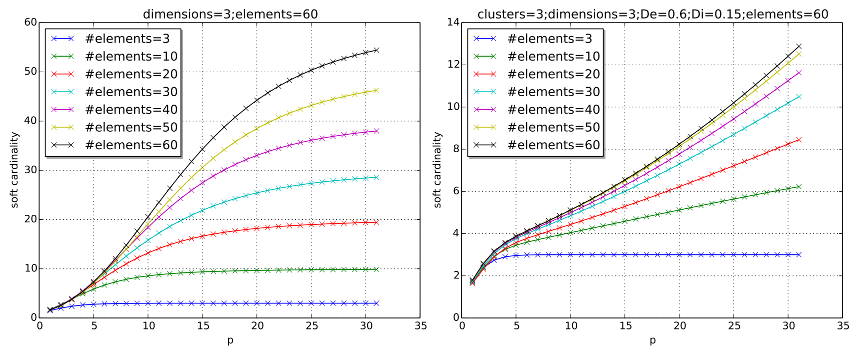


Figure 3.7: Soft cardinalities of different sized subsets varying the softness control parameter p (uniform data at left and clustered at right)

3.8 Conclusion

In this paper, we have shown that the soft cardinality approach has: plausible intuitions, theoretical foundation, experimental validity, practical usefulness and good performance. Regarding the intuitions about soft counting, the presented model correctly put the idea of a cardinality measure in a setting where similarities among elements reveal and remove redundancy. In addition, the proposed theoretical definition and mathematical properties provide a solid foundation for the model. Experimental validation conducted using synthetic data corroborated the main intuition and properties of the soft cardinality. Besides, the fact that the soft cardinality can be used analogously and interchangeably with the classical cardinality allows the use of a wide variety of available methods and techniques.

Chapter 4

Soft cardinality in semantic text processing:

Experience of the SemEval international competitions

Soft cardinality is a generalization of the classic set cardinality (i.e. the number of elements in a set), which exploits similarities between elements to provide a "soft" counting of the number of elements in a collection. This model is so general that can be used interchangeability as cardinality function in resemblance coefficients such as Jaccard's, Dice's, cosine and others. Beyond that, cardinality-based features can be extracted from pairs of objects being compared to learn adaptive similarity functions from training data. This approach can be used for comparing any object that can be represented as a set or bag. We and other international teams used soft cardinality to address a series of natural language processing (NLP) tasks in the recent SemEval (semantic evaluation) competitions from 2012 to 2014. The systems based on soft cardinality have always been among the best systems in all the tasks in which they participated. This paper describes our experience in that journey by presenting the generalities of the model and some practical techniques for using soft cardinality for NLP problems.

A version of this chapter was accpeted for publication at *Polibits* journal (May 2015).

4.1 Introduction

SemEval[1] (Semantic Evaluation) is a series of academic workshops which aims to bring together the scientific community in the field of natural language processing (NLP) around tasks involving automatic analysis of texts. Each year, a set of challenges is proposed dealing with different aspects of the area of computational semantics attracting the attention of research groups of institutions worldwide. Each challenge follows a peer reviewing screening process ensuring the relevance, correctness, quality, and fairness of each competition. Task organizers pose an interesting challenge by providing a new dataset and a methodology for evaluating systems that address that challenge. For instance, organizers of the semantic textual similarity task (STS) provide several training datasets containing pairs of short texts labeled with a gold standard built using human annotators. Next, participating teams build systems that predict annotations in unseen test data, and organizers evaluate the performance of each system. Finally, organizers and participants describe their experiences and used approaches in peer-reviewed articles, which become de facto state of the art methods.

We, researchers from the Universidad Nacional de Colombia and the Centro de Investigación en Computación of the IPN in Mexico collaborated to participate in several SemEval tasks since 2012. The core component of our participating systems is soft cardinality [**?**], a recently proposed approach to make the classic cardinality of set theory sensitive to the similarities and differences between the elements in a collection. This approach is particularly appropriate for addressing NLP problems because it allows finding commonalities between texts that do not share words but have words that are similar in some degree. Somehow surprisingly, systems build with this simple approach obtained impressive results in several SemEval challenges defeating considerably more complex and costly approaches. In addition, our team was the one with the highest number of participations from 2012 to 2014 also using the same core approach for addressing all tasks and always obtaining very satisfactory results.

This paper describes our experience in that journey by reviewing our participations in SemEval. Section 4.2 presents a brief description of soft cardinality, some parameterized resemblance coefficients, and the method for extracting cardinality-based feature representations. Section 4.3 presents some of the techniques and resources used for addressing NLP tasks using soft cardinality. Section 4.4 reviews the systems and particular tasks addressed in SemEval, and a summary of the obtained results is presented. Finally in Section 4.5 we provide some concluding remarks.

4.2 Soft cardinality approach

The cardinality of a collection of elements is the counting of non-repeated elements within. This definition is intrinsically associated with the notion of set,

[1]http://en.wikipedia.org/wiki/SemEval

which is a collection of non-repeating elements. The notation of the cardinality of a collection or set A is $|A|$. Jimenez et al. [**?**] proposed *soft cardinality*, which uses a notion of similarity between elements for grouping not only identical elements but similar too. That notion of similarity between elements is provided by a similarity function that compares two elements a_i and a_j returning a score in [0,1] interval having $sim(x, x) = 1$. Although, it is not necessary that sim fulfills another mathematical property aside identity, symmetry is also desirable. Thus, the soft cardinality of a collection A, whose elements $a_1, a_2, \ldots, a_{|A|}$ are comparable with a similarity function $sim(a_i, a_j)$, is denoted as $|A|_{sim}$. This soft cardinality is given by the following expression:

$$|A|_{sim} = \sum_{i=1}^{|A|} \frac{w_{a_i}}{\sum_{j=1}^{|A|} sim(a_i, a_j)^p} \tag{4.1}$$

It is trivial to see that $|A| = |A|_{sim}$ if either $p \to \infty$ or when the function sim is a crisp comparator, i.e. one that returns 1 for identical elements and 0 otherwise. This property shows that soft cardinality generalizes classic cardinality and that the parameter p controls its degree of "softness", the default value is $p = 1$. The values w_{a_i} are optional "importance" weights associated with each element a_i, by default those weights can be assigned to 1.

4.2.1 Inferring intersection cardinality

The soft cardinality of the intersection of two collections cannot be calculated directly from $A \cap B$ because the intersection operator is inherently crisp. This means that, if there are no common elements between A and B, their intersection is empty, and so its soft cardinality is 0. The following definition allows inferring the soft cardinality of the intersection through soft cardinalities of each collection and their union.

Let A and B be two collections, the soft cardinality of their intersection is $|A \cap B|_{sim} = |A|_{sim} + |B|_{sim} - |A \cup B|_{sim}$. In this case, the operator $\cup$ means ***bag union***, which takes the maximum number of occurrences of the elements in each bag. Example: $\{1, 1, 2, 3\} \cup \{1, 2, 2\} = \{1, 1, 2, 2, 3\}$[**?**].

This infers non-empty intersections for pairs of collections that have not common elements, but have similar elements. Once $|A \cup B|_{sim}$, $|A \cap B|_{sim}$, $|A|_{sim}$ and $|B|_{sim}$ are known, it is possible to obtain all other areas in the Venn's diagram of two sets, i.e. $|A \triangle B|_{sim} = |A \cup B|_{sim} - |A \cap B|_{sim}$, $|A \setminus B|_{sim} = |A|_{sim} - |A \cap B|_{sim}$ and $|B \setminus A|_{sim} = |B|_{sim} - |A \cap B|_{sim}$. These are the building blocks of almost any cardinality-based resemblance coefficient.

4.2.2 Cardinality-based resemblance coefficients

Since more than a century when Jaccard [**?**] proposed his well-known index, the classic set cardinality has been used to build similarity functions for set comparison. Basically, any cardinality-based similarity function is an algebraic combination of $|A|$, $|B|$ and either $|A \cap B|$ or $|A \cup B|$ (e.g. Jaccard, Dice [**?**],

Table 4.1: Named Resemblance Coefficients

Resemblance coefficient	$SIM(A,B) =$
Jaccard [?]	$\frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$
Dice or Sørensen [?]	$\frac{\lvert A\cap B\rvert}{0.5(\lvert A\rvert+\lvert B\rvert)}$
Overlap	$\frac{\lvert A\cap B\rvert}{\min(\lvert A\rvert,\lvert B\rvert)}$
Cosine or Ochiai [?]	$\frac{\lvert A\cap B\rvert}{\sqrt{\lvert A\rvert\cdot\lvert B\rvert}}$
Hamming	$\frac{1}{1+\lvert A\triangle B\rvert}$

Tversky [?], overlap and cosine [?] coefficients). Table 4.1 shows some of the most used resemblance coefficients.

The simplest way to build similarity functions with soft cardinality is to replace the classic cardinality $|*|$ by soft cardinality $|*|_{sim}$. These coefficients have mathematical properties (e.g. transitivity, metric properties) that make of them a good option for many applications. When cosine coefficient is used in combination with soft cardinality, the resulting approach is conceptually similar to the soft cosine measure proposed by Sidorov et al. [?].

4.2.3 Parameterized resemblance coefficients

Some resemblance coefficients contain in its formulation parameters that allow adaptation to particular tasks. One of them is the Tversky's index [?], which was proposed as a cognitive model of similarity:

$$\begin{aligned} SIM(A,B) &= \frac{|A\cap B|}{\alpha|A\setminus B|+\beta|B\setminus A|+|A\cap B|}; \\ \alpha,\beta &\geq 0 \end{aligned}$$

There, parameters α and β control the balance of the differences between A and B. In Tversky's model, one of the sets being compared is the *referent* and the other is the *variant*, making this similarity measure asymmetric when $\alpha \neq \beta$. This asymmetry makes of Tversky's model an inclusion measure rather than a similarity measure. Nevertheless, in its original form it is still useful in text applications where the texts being compared have an ordinal relation, e.g. question-answer in question answering, query-document in information retrieval, text-hypothesis in textual entailment, text-summary in summarization, and others. In applications such as textual similarity or paraphrase detection, symmetry plays an important role. Jimenez et al.[?] proposed a symmetrization of Tversky's index in the following way:

$$SIM(A,B) = \frac{c}{\beta(\alpha a + (1-\alpha)b) + c} \tag{4.2}$$

$$|c| = |A\cap B| + bias,$$

$$a = \min[|A \setminus B|, |B \setminus A|],$$

$$b = \max[|A \setminus B|, |B \setminus A|].$$

This formulation also re-arranges parameters α and β in a way that α controls the balance between the differences of A and B, and β controls the importance in the denominator between differences and commonalities between A and B. The additional parameter *bias* allows removing an implicit degree of similarity between A and B, so usually $bias \leq 0$. This parameter can also be associated with the average or minimum intersections in a dataset. This coefficient generalizes Jaccard ($\alpha = \beta = 1; bias = 0$), Dice ($\alpha = \beta = 0.5; bias = 0$), overlap ($\alpha = 1; \beta = 0; bias = 0$) and Hamming ($\alpha = 1; \beta = 1; bias = 1 - |A \cap B|$).

Another generalization can be made by the observation that Dice and cosine coefficients are the ratio of $|A \cap B|$ and the arithmetic and geometric means, respectively. Therefore, the denominator can be replaced the expression of the generalized mean between $|A|$ and $|B|$:

$$SIM_p(A, B) = \frac{|A \cap B|}{0.5(|A|^p + |B|^p)^{1/p}} \tag{4.3}$$

Different values of the parameter p produce different known coefficients, i.e. Dice ($p = 1$), cosine ($p \to 0$) and overlap ($p \to \infty$). Other interesting values of p correspond to known means: $p = -1$ is the harmonic mean, $p = 2$ is the quadratic mean and $p \to -\infty$ is the minimum.

De-Baets and De-Meyer [?] proposed another hexaparametric generalized resemblance coefficient (a and b as in Eq. 4.2:

$$SIM(A, B) = \frac{\alpha a + \beta b + \delta |A \cap B|}{\alpha' a + \beta' b + \delta' |A \cap B|}$$

The values selected for parameters in resemblance coefficients are usually obtained by optimizing some criterion using training data. For example, in a dataset that consist of triples (A, B, g_{AB}) where g_{AB} is a gold standard of similarity (e.g. agreement of human judgments), the optimal set of parameters can be obtained by maximizing the correlation (Pearson or Spearman) between $SIM(A, B)$ and g_{AB} or by minimizing the mean-absolute error (MAE) or root-mean-squared error (RMSE).

4.2.4 Cardinality-based features for machine learning models

The parameterized resemblance coefficients allow the exploration and adaptation of a relatively large set of similarity functions to a particular problem. However, the space of possible formulations of similarity functions is huge. Which is the most appropriate similarity function for a particular problem is a question that can be addressed by adjusting parameters in these coefficients, but this strategy is nothing more than an arbitrary bias in the search. In this case, “a problem" means a dataset that needs to be modeling or explained by the similarity function. An exhaustive exploration of candidate similarity function is out of question given the large number of possible formulations. Genetic programming [**?**] can be used for this, but still the considered functions might be unable to model local non-linearities in some datasets. Using machine learning methods may be an appropriate option to address these issues.

Most machine learning algorithms builds models using a fixed features set (i.e. a vector) to represent each sample (e.g. linear regression, support vector machines, naïve Bayes, decision trees, K-means, etc.) Training data is a set of samples wherein each sample is associated with a target variable, a similarity score in our scenario. These labeled samples are used to construct a black box model, which is able, to some extent, to predict the target variable, and it is also able of producing predictions for unlabeled data. There is a variety of methods for obtaining these black box models including approaches whether geometric, probabilistic, algorithmic, information theoretical, among many others. This approach allows learning a similarity function adapted to the problem at hand efficiently and generally with a good level of generalization.

The proposed approach consists in extracting a fixed set of features from each pair of sample objects A and B, building a training dataset using these features, and labeling each sample with a gold-standard of similarity. Next, this training dataset is used to learn a machine learning model for the target variable. Finally, the learned model is used to provide similarity scores for other pairs of objects by extracting from them the same features set.

The proposed features for each pair of objects are based on cardinality, using either classical or soft cardinality. Thus, for a pair of objects (A, B) represented as sets (or bags), the basic set of cardinality-based features consist of $|A|$, $|B|$ and $|A \cup B|$. All other possible cardinality-based features are mathematical combinations thereof these three features. The following obvious features are the other areas in the Venn's diagram of two sets, i.e. $|A \cap B|$, $|A \bigtriangleup B|$, $|A \setminus B|$ and $|B \setminus A|$, Table 4.2 shows the basic and derived set of features described. An additional set of features aimed to enable machine learning algorithms to identify symmetrical patterns in the objects being compared is built using min() and max() functions, see "Derived set 2" in Table 4.2.

Although, many machine learning methods requires or includes previous preprocessing steps of normalization or standardization of the features. Therefore, it makes sense to produce some features whose values are limited in a range. Table 4.3 shows an extended set of features limited to [0,1] interval. These features are aimed to allow machine learning algorithms for learning patterns from the relative proportions of cardinality magnitudes. In the context of text applications, these rational features allow identifying patterns that are independent of the length of texts.

4.2.5 Exploring larger sets of features

Feature sets presented in the previous section have shown effective to address many natural language processing challenges at SemEval competitions. Despite

Table 4.2: The basic and derived feature sets for the comparison two collections of words.

Basic	**Derived set 1**
$\|A\|$	$\|A \cap B\| = \|A\| + \|B\| - \|A \cup B\|$
$\|B\|$	$\|A \triangle B\| = \|A \cup B\| - \|A \cap B\|$
$\|A \cup B\|$	$\|A \setminus B\| = \|A\| - \|A \cap B\|$
	$\|B \setminus A\| = \|B\| - \|A \cap B\|$

Derived set 2
$\max(\|A\|, \|B\|)$
$\min(\|A\|, \|B\|)$
$\max(\|A \setminus B\|, \|B \setminus A\|)$
$\min(\|A \setminus B\|, \|B \setminus A\|)$

Table 4.3: Set of ten extended rational features.

	Feature expression		**Feature expression**
#1	$^{\|A\|}/_{\|A \cup B\|}$	#6	$^{\|B\|-\|A \cap B\|}/_{\|B\|}$
#2	$^{\|A\|-\|A \cap B\|}/_{\|A\|}$	#7	$^{\|B\|-\|A \cap B\|}/_{\|A \cup B\|}$
#3	$^{\|A\|-\|A \cap B\|}/_{\|A \cup B\|}$	#8	$^{\|A \cap B\|}/_{\|B\|}$
#4	$^{\|A \cap B\|}/_{\|A\|}$	#9	$^{\|A \cap B\|}/_{\|A \cup B\|}$
#5	$^{\|B\|}/_{\|A \cup B\|}$	#10	$^{\|A \cup B\|-\|A \cap B\|}/_{\|A \cup B\|}$

their effectiveness, they seem to be arbitrary. For example, features shown in Table 4.3 are rational combinations of some of the features in Table 4.2. Why only select these ten combinations? In fact, if the Table 4.2 contains 11 features and number 1 is added to this set, then the number of possible combinations of rational features is $12 \times 11 = 131$. With this new set of 131 features, the ten features in Table 4.3 seems to be arbitrary indeed. The reason for including number 1 in the basic feature set is thereby allowing the basic features and their inverses be included in the combined feature set, e.g. $|A \triangle B|$ and $^1/_{|A \triangle B|}$. Note that Jaccard index (i.e. $^{|A \cap B|}/_{|A \cup B|}$) is also included in this combined set. Let's call the basic set of features F, formally:

$$\begin{aligned} F(A,B) &= \{1, |A|, |B|, |A \cup B|, |A \cap B|, |A \triangle B|, \\ &\quad |A \setminus B|, |B \setminus A|, \min(|A|,|B|), \max(|A|,|B|), \\ &\quad \min(|A \setminus B|, ||B \setminus A|), \max(|A \setminus B|, ||B \setminus A|)\} \end{aligned}$$

Before providing a formal definition of the combined set of features, an additional set of basic features from different means (averages) must be considered as well. These additional features allow include Dice, cosine, and other coefficients as features too. For that, the expression of the generalized mean (see denominator at Eq. 4.3) can be used considering only a representative subset of the possible values for parameter p:

$$\begin{aligned} P &= \{-50, -20, -10, -4, -3, -2, -1, \\ &\quad 0.0001, 1, 2, 3, 4, 10, 20, 50\} \end{aligned}$$

Now, the basic feature set F can be extended to F' by including all the generalized means restricted by P, between $|A|$ and $|B|$, and between $|A \setminus B|$ and $|B \setminus A|$, formally :

$$\begin{aligned} F'(A,B) &= F(A,B) \cup \\ &\quad \{0.5(|A|^p + |B|^p)^{1/p} \,|\forall p \in P\} \cup \\ &\quad \{0.5(|A \setminus B|^p + |B \setminus A|^p)^{1/p} \,|\forall p \in P\} \end{aligned}$$

Now, the number of features in $F'(A,B)$ is $|F(A,B)| + 2|P| = 42$ features. The combined set of features can be defined as:

$$C(A,B) = \left\{ \frac{f_1}{f_2} \,|(f_1, f_2) \in F'(A,B) \times F'(A,B) \wedge f_1 \neq f_2 \right\}$$

This combination produce $42 \times 41 = 1,722$ features in $C(A,B)$. This is a very large number of features for comparing only two set. Clearly, only subsets of this set of features are useful for particular applications. Even different datasets for a same task could require different representations. The idea is to make a selection of features (see [?] for an introductory tutorial) before using any machine learning regressor or classifier for a particular task. This allows to learn an adequate representation for the task prior to learn and adequate black-box (or even an interpretable) model for addressing the task. The optimal feature set for a particular task is very difficult to find because it would require considering $2^{1,772}$ possible subsets. Generally, using known methods only a near-optimal subset can be found, whose size is usually not too small nor too large. Jimenez et al. [?] observed that as a general rule the number of near-optimal features is between 10% and 20% of the number of available training samples. However, the larger the number of possible features explored, the higher the chances of finding smaller feature subsets. For example, Dueñas et al. [?] considering a similar feature set but also including logarithmic functions, found that the most correlated feature to the difficulty of a short-answer question was $\frac{|A \setminus B|}{\log(0.5\sqrt{|A|^2+|B|^2})}$, where A corresponds to the text of the reference answer and B to the question.

Although, we did not use these cardinality-based feature representation learned from training data in SemEval competitions, in subsequent studies showed this approach effective for lexical similarity task and in the analysis of questions for student evaluation. Therefore, we believe this approach may also be useful for other applications of NLP.

4.3 Using soft cardinality for NLP

4.3.1 Textual similarity

The way to build a text similarity function is *i)* to select a linguistic unit to be compared (e.g. sentences), *ii)* to use a representation of the texts based in bags (e.g. bags of words, n-grams, dependencies, etc.), *iii)* to choose a cardinality based similarity coefficient (e.g. Jaccard's, Tversky's, De Beat's coefficients), and *iv)* to provide a pairwise similarity function SIM_{word} for comparing the elements produced by the used text representation (e.g. normalized Levenshtein similarity, nPMI [**?**], normalized path length in WordNet [**?**], etc.). The simplest example of such similarity function for sentence pairs is:

$$SIM_{sentence}(A,B) = \frac{|A \cap B|_{SIM_{word}}}{|A \cup B|_{SIM_{word}}}. \tag{4.4}$$

The only parameter to be adjusted in Eq. 4.4 is p, the softness controller parameter. Jimenez et al. [**?**] showed that the default $p = 1$ works well for short sentences in English. However, a suitable value for p depends primarily on the range and distribution of the values returned by SIM_{word}, on the length of the texts, and on the task at hand. Clearly, any resemblance coefficient presented in Section 4.2.2 and Section 4.2.3 can be used.

It is important to note that Eq. 4.4 is recursive, similar to the popular Monge-Elkan measure [**? ?**]. That is, the similarity function $SIM_{sentence}$ is obtained from another similarity function, SIM_{word}. This idea can be recursively used to build a similarity function $SIM_{paragraph}$ based on $SIM_{sentence}$, and so on. Thus, it is possible to build similarity functions exploiting the hierarchical structure of the text and natural language.

4.3.2 Term weights

Term weighting is a common practice in NLP to promote informative words and ignore non-informative words. For instance, the so-called stopwords are function words that can be removed of texts preserving their meaning to some extent, examples of these stopwords are *the*, *of*, *for*, etc. Removing stopwords may be interpreted as a binary weighting for the words in a text, i.e. assigning 1 for non-stopwords and 0 otherwise. However, a graded notion of informativeness has proven more effective than the binary approach. Probably the most used term-weighting schemes are tf.idf [**?**] and BM25 [**?**].

The soft cardinality allows the use of term weights at w_{a_i} in Eq. 4.1. It is important to note, that elements with zero weights (or close to 0) should be removed from texts because, although their contribution is 0, their similarities still interacts with the other elements affecting soft cardinality. This issue reveals the fact that most of the properties of the soft cardinality get overwritten because of term weighting. However, that weighted approach still preserves the original motivations of soft cardinality and extends its modeling capability [**?**].

4.3.3 Features for text comparison

In Section 4.2.4 we presented a method for extracting basic sets of cardinality-based features from a pair of texts represented as sets or bags of words. When the soft cardinality is being used in short texts, its word-to-word similarity function SIM_{word} plays a central role in the meaning of the extracted features. For instance, if the SIM_{word} compares words morphologically, then features extracted using $|*|_{SIM_{word}}$ reflect morphological features in texts. Additionally, other types of features can be extracted by modifying the set representation of text. For instance, a text A can be enriched with words taken from the dictionary definitions of the words already in A. These and others methods for feature extraction are presented in the following sections.

4.3.3.1 Morphological features

For extracting morphological features of texts it is only necessary to provide a $SIM_{word}(w_1, w_2)$ function based on the characters of the words. Some options are edit distance [?] (converted to a similarity function) or Jaro-Winkler similarity [?] (see [?] for a survey). Our choice was to use the Tversky symmetrized index (Eq. 4.2) by representing each word by 3-grams of characters, e.g. *house* is represented as $\{hou, ous, use\}$. The values of the parameters of the Tversky symmetrized index were obtained by building a simple text similarity function $SIM_{sentence}(A, B)$ using Dice's coefficient and soft cardinality using that function as auxiliary similarity function, i.e. $|*|_{SIM_{word}}$by Eq. 4.1. Then the space of parameters were explored by hill-climbing optimizing the Pearson's correlation between the similarity score obtained $SIM_{sentence}$ and the gold standard of the SICK dataset [?]. The optimal values of the parameters were $\alpha = 1.9$, $\beta = 2.36$, $bias = -0.97$. In fact, the size of n-grams, $n = 3$, was also optimal for that function. The softness-control parameter of soft cardinality was optimized too, obtaining $p = 0.39$, but it is irrelevant for SIM_{word}. Thus, the proposed similarity function for comparing words is:

$$\begin{aligned} SIM_{word}(w_1, w_2) &= \frac{|w_1 \cap w_2| - 0.97}{2.36(1.9a - 0.9b) + |w_1 \cap w_2| - 0.97} \\ a &= \min[|w_1 \setminus w_2|, |w_2 \setminus w_1|] \\ b &= \max[|w_1 \setminus w_2|, |w_2 \setminus w_1|] \end{aligned} \tag{4.5}$$

Finally, having soft cardinality $|*|_{SIM_{word}}$ for each pair of texts A and B the features described in Section 4.2.4 or Section 4.2.5 can be obtained straightforwardly.

4.3.3.2 Semantic features

The proposed $SIM_{word}(w_1, w_2)$ function in previous section only exploits the superficial information of the words, therefore the extracted features using soft cardinality $|*|_{SIM_{word}}$ convey the same kind of information but at textual level. The obvious next step is to use a function of similarity of words that exploits semantic relationships between the words instead of comparing letters. In that way, the soft cardinality-based features would convey semantic information. There are several choices for that. First, knowledge-based lexical measures based on WordNet can do the job (see background section in [?].) Alternatively, distributional representations that make use of frequencies of the words taken from large corpora (see [?] for some examples) can be used for semantic lexical similarity. Recently, neural word embedding [? ?] has become the state-of-the-art for semantic lexical similarity. The approach consists in building a predictive model for each word in the vocabulary of a large corpus based in local contexts. For this, each vocabulary word is represented as a fixed dimensional vector (usually from 100 to 300 dimensions). These representations are those that maximize the probability of generating the entire corpus. Although, the process of obtaining these representations is computationally expensive, pre-trained vectors are freely-available for use [2] [3]. To obtain similarity scores with this approach, the cosine similarity between their vectorial representations is used.

4.3.3.3 ESA Features

For this set of features, we used the idea proposed by Gabrilovich and Markovitch [**?**] of extending the representation of a text by representing each word by its textual definition in a knowledge base, i.e. explicit semantic analysis (ESA). For that, we used as knowledge base the synset's textual definitions provided by WordNet. First, in order to determine the textual definition associated to each word, the texts were tagged using the maximum entropy POS tagger included in the NLTK[4]. Next, the Adapted Lesk's algorithm [**?**] for word sense disambiguation was applied in the texts disambiguating one word at the time. The software package used for this disambiguation process was *pywsd*[5]. The argument parameters needed for the disambiguation of each word are the POS tag of the target word and the entire sentence as context. Once all the words are disambiguated with their corresponding WordNet synsets, each word is replaced by all the words in their textual definition jointly with the same word and its lemma. The final result of this stage is that each text in the dataset is replaced by a longer text including the original text and some related words. The motivation of this procedure is that the extended versions of each pair of texts have more chance of sharing common words that the original texts.

Once the extended versions of the texts were available, the same features described in Section 4.3.3.1 or Section 4.3.3.2 can be obtained.

4.3.3.4 Features for each part-of-speech category

This set of features is motivated by the idea proposed by Corley and Mihalcea [**?**] of grouping words by their POS category before being compared for semantic textual similarity. Our approach provides a version of each text pair in the dataset for each POS category including only the words belonging to that category. For instance, the pair of texts {"*A beautiful girl is playing tennis*", "*A nice and handsome boy is playing football*"} produces new pairs such as: {"*beautiful*", "*nice handsome*"} for the ADJ tag, {"*girl tennis*", "*boy football*"} for NOUN and {"*is playing*", "*is playing*"} for VERB.

Again, the POS tags were provided by the NLTK's maximum entropy tagger. The 28 POS categories were simplified to nine categories in order to avoid an excessive number of features and hence sparseness; used mapping is shown in Table 4.4. Next, for each one of the nine new POS categories a set of features is

[2]http://code.google.com/p/word2vec/
[3]http://nlp.stanford.edu/projects/glove/
[4]http://www.nltk.org/
[5]https://github.com/alvations/pywsd

Reduced tag set	**NLTK's POS tag set**
ADJ	JJ,JJR,JJS
NOUN	NN,NNP,NNPS,NNS
ADV	RB,RBR,RBS,WRB
VERB	VB,VBD,VBG,VBN,VBP,VBZ
PRO	WP,WP$,PRP,PRP$
PREP	RP,IN
DET	PDT,DT,WDT
EX	EX
CC	CC

Table 4.4: Mapping reduction of the POS tag set

extracted reusing again the method proposed in section 4.2.4. The only difference consideration is the stopwords should not be removed and stemming should not be performed. The motivation for generating this feature sets grouped by POS category is that the machine learning algorithms could weight differently each category. The intuition behind this is that it is reasonable that categories such as VERB and NOUN could play a more important role for the task at hand than others such as ADV or PREP. Using these categorized features, such discrimination among POS categories can be discovered from the training data.

4.3.3.5 Features from dependencies

Syntactic soft cardinality [**?** **?**] extends the soft cardinality approach by representing texts as bags of dependencies instead of bags of words. Each dependency is a 3-tuple composed of two syntactically related words and the type of their relationship. For instance, the sentence "*The boy plays football*" is be represented with 3 dependencies: [**det**,"*boy*","*The*"], [**subj**,"*plays*","*boy*"] and [**obj**,"*plays*","*football*"]. Clearly, this representation distinguishes pairs of texts such as {"*The dog bites a boy*","*The boy bites a* dog"}, which are indistinguishable when they are represented as bags of words. This representation can be obtained automatically using the Stanford Parser [**?**], which, in addition, provides a dependency identifying the root word in a sentence.

Once the texts are represented as bags of dependencies, it is necessary to provide a similarity function between two dependency tuples in order to use soft cardinality, and hence to obtain the cardinality-based features in Table 4.2. Such function can be obtained using the SIM_{word} function (Eq. 4.5) for comparing the first and second words between the dependencies and even the labels of the dependency types. Let's consider two dependencies tuples $d = [d_{dep}, d_{w_1}, d_{w_2}]$ and $p = [p_{dep}, p_{w_1}, p_{w_2}]$ where d_{dep} and p_{dep} are the labels of the dependency type; d_{w_1} and p_{w_1} are the first words on each dependency tuple; and d_{w_2} and p_{w_2} are the second words. The similarity function for comparing two dependency tuples can be a linear combination of the sim scores between the corresponding elements of the dependency tuples by the following expression:

$$sim_{dep}(d,p) = \gamma sim(d_{dep}, p_{dep}) + \delta sim(d_{w_1}, p_{w_2}) + \lambda sim(d_{w_2}, p_{w_2})$$

Although, it is unusual to compare the dependencies' type labels d_{dep} and p_{dep} with a similarity function designed for words, we observed experimentally that this approach yield better overall performance in the textual relatedness task in comparison with a simple exact comparison. The optimal values for the parameters $\gamma = -3$, $\delta = 10$ and $\lambda = 3$ were determined with the same methodology used in Section 4.2.3 for determining α, β and $bias$. Clearly, the fact that $\delta > \lambda$ means that the first words in the dependency tuples plays a more important role than the second ones. However, the fact that $\gamma < 0$ is counter intuitive because it means that the lower the similarity between the dependency type labels is, the larger the similarity between the two dependencies. Up to date, we have been unable to find a plausible explanation for this phenomenon.

4.4 Soft cardinality at SemEval

The soft cardinality approach has been used by several teams for participating in several tasks in the recent SemEval campaigns (2012 to 2014). In SemEval, the task organizers propose a NLP task, provide datasets, and an evaluation setup that is carried out by them. This methodology ensures a fair comparison of the performance of the methods used by competitors. The participating systems that incorporated soft cardinality among their used methods have obtained very satisfactory results, obtaining in most of the cases rankings among the top systems. In this section, a brief overview of these participations is presented.

4.4.1 Semantic textual similarity

The task of automatically comparing the similarity or relatedness between pairs of texts is fundamental in NLP, which attracted the attention of many researchers in the last decade [**?** **?**]. This task consists in building a system able to compare pairs of texts, using (or not) training data and return graded predictions of similarity or relatedness. The system performance is evaluated by correlating its predictions against a gold standard built using human judgments in a graded scale. Table 4.5 contains a summary of the results obtained by the systems that used soft cardinality.

In 2012, soft cardinality was used for the first time [**?**] in the pilot of the Semantic Textual Similarity (STS) task [**?**]. The approach consisted in building a cardinality-based similarity function $SIM_{sentence}$ combining soft cardinality with a coefficient similar to Tversky's (see Subsection 4.3.1.) The function SIM_{word} used for comparing pairs of words was based on n-grams of characters combined with the same rational coefficient used at sentence level (see Eq. 4.5.) The parameters p, n and those of both coefficients were obtained by looking for an optimal combination in the provided training data. Finally, tf-idf weights were associated with the words (weights w_{a_i} in Eq. 4.1.) This simple approach obtained an unexpected third place in the official ranking among 89 participating systems. Besides, as Table 4.5 shows, this system was pretty close to the top system, which used considerably more resources [?]. Besides, comparing the rankings obtained for individual datasets and the overall ranking (3^{rd}), it can be seen that the soft cardinality system was more consistent across different data sets than most of the other systems.

In 2013, the STS task was proposed again but with increased difficulty because no additional data was provided for training. Our 2012 approach was extended by building an additional similarity function for sentences using nPMI [?] as the comparator of words. Moreover, the predictions were obtained training a regression SVM with the features described in Subsection 4.2.4. This system ranked 19^{th} among 89 systems. However, in addition to the official results, we discovered that the same 2012 function averaged with the new nPMI function correlated much better (4^{th}) [?].

In addition, in 2013, a pilot for the Typed Similarity task was proposed. It consisted in comparing pairs of text records associated with objects from the Europeana[6] database. Croce et al. [?] built a system based on the previously proposed *syntactic soft cardinality* [?]. This consists in representing texts as sets of triples (*word1*, *word2*, *relation*) extracted from dependency graphs, and combine them using soft cardinality with a similarity function for those triplets. This system ranked first among 15 participants.

In 2014, the task 10 at SemEval was the third STS version [?], which included additional datasets in Spanish. Lynum et al. [?] proposed a system for the data sets in English using features (among others) extracted with soft cardinality ranking first in 4 out of 6 data sets among 37 participating systems. Similarly, Jimenez et al. [?] proposed a system based on the soft cardinality for the Spanish data sets, ranking first in one of the data sets and third overall among 22 systems. This system also participated in tasks 1 [?] and 3 [?], which addressed text relatedness and similarity between different lexical levels (e.g. paragraph to sentence) respectively. In these tasks, the systems based on the soft cardinality ranked 4^{th} out of 17, and 3^{rd} out of 38 systems. The used features were a combination of the feature sets presented in sections 4.3.3.1, 4.3.3.2, 4.3.3.3, 4.3.3.3, 4.3.3.4, and 4.3.3.5.

These results show that soft cardinality is a very competitive tool for building text similarity functions with relatively few resources, namely: a similarity function for comparing pairs of words, soft cardinality, and a cardinality-based coefficient or a regression method to learn this coefficient.

4.4.2 Textual Entailment

Textual entailment (TE) is the task that consists in determining whether or not a text entails another one. It was proposed under the name *cross-lingual textual entailment* (CLTE) [? ?] in SemEval with the additional difficulty of having the two texts in different languages. The results obtained by the systems based on the soft cardinality that participated in this task in 2012 and

[6]http://www.europeana.eu/

Table 4.5: Best results obtained by systems that used soft cardinality in SemEval 2012-2014 for textual similarity and relatedness (Pearson correlation)

Year	Task	Dataset	Rank	Soft Card.†	Top Sys.‡	Ref.
2012	STS	MSRpar	7th/89	0.6405	0.7343	[?]
		MSRvid	9th/89	0.8562	0.8803	
		SMT-eur	9th/89	0.5152	0.5666	
		OnWN	3rd/89	0.7109	0.7273	
		SMT-news	11th/89	0.4833	0.6085	
		All (w. mean)	**3rd/89**	**0.6708**	**0.6773**	
2013	STS	Headlines	30th/90	0.6713	0.7838	[?]
		OnWN	7th/90	0.7412	0.8431	
		FNWM	22th/90	0.3838	0.5818	
		SMT	54th/90	0.3035	0.4035	
		All (w. mean)	**19th/90**	**0.5402**	**0.6181**	
		All (unofficial)	**4th/90**	**0.5747**	**0.6181**	
	Typed sim.	**Europeana**	**1st/15**	**0.7620**	**0.7620**	[?]
2014	Task 1-STS	**SICK**	**4th/17**	**0.8043**	**0.8280**	[?]
	Task 3	Para2Sent	1st/38	0.8370	0.837	
		Sent2Phr	6th/38	0.7390	0.7770	
		Phr2Word	3rd/22	0.2740	0.4150	
		Word2Sense	5th/20	0.2560	0.3890	
		All (w. mean)	**3rd/38**	**0.5260**	**0.5810**	
	Task 10 (en)	deft-forum	1st/38	0.5305	0.5305	[?]
		deft-news	2nd/37	0.7813	0.7850	
		headlines	1st/37	0.7837	0.7837	
		images	1st/37	0.8343	0.8343	
		OnWN	4th/37	0.8502	0.8745	
		tweet-news	1st/37	0.7921	0.7921	
		All (w. mean)	**3rd/38**	**0.7549**	**0.7610**	
	Task 10 (es)	Wikipedia	1st/22	0.7804	0.7804	[?]
		news	7th/22	0.8154	0.8454	
		All (w. mean)	**3rd/22**	**0.8013**	**0.8072**	

†Results for the best system using the soft cardinality

‡Results for the best system in competition

Table 4.6: Best results for accuracy obtained by the systems that used the soft cardinality in SemEval 2012 to 2014 for the cross-lingual textual entailment task (CLTE)

Year	Dataset	Rank	Soft Card.	Top Sys.	Ref.
2012	Spanish-English	$5^{th}/29$	0.552	0.632	[?]
	Italian-English	$1^{st}/21$	0.566	0.566	
	French-English	$1^{st}/21$	0.570	0.570	
	German-English	$3^{rd}/21$	0.550	0.558	
2013	Spanish-English	$1^{st}/15$	0.434	0.434	[?]
	Italian-English	$1^{st}/15$	0.454	0.454	
	French-English	$6^{th}/15$	0.426	0.458	
	German-English	$6^{th}/16$	0.414	0.452	

2013 are shown in Table 4.6. The approach consisted in providing two versions of the pair of texts, each one in a single language, using machine translations from Google translate[7]. Once in a single language, the soft cardinality features explained in Subsection 4.2.4 were extracted for each text pair using the same word-to-word similarity function SIM_{word} used for STS. Finally, these features were combined by a classifier to determine the type of entailment. In 2013, Jimenez et al. [?] showed that these features are also language independent, making possible to train a single classifier using data in different languages. This approach produced (not included in the official ranking) state-of-the-art results for all CLTE datasets [?].

In 2014, the textual entailment task was proposed for the SICK dataset (Sentences Involving Compositional Knowledge) [?]. Using the same approach as in CLTE, but combining additional features from soft cardinalities obtained with word similarity functions based on WordNet, ESA and dependency graphs, the soft-cardinality system [?] ranked 3^{rd} among 18. Table 4.7 shows the results obtained by the soft cardinality system both in textual entailment and textual relatedness sub-tasks.

4.4.3 Automatic students' answer grading

The task consisted in grading the correctness of a student answer (SA) to a question (Q) given a reference answer (RA) [?]. The approach of the system that used soft cardinality [?] consisted in extracting features for pairs (SA,Q), (Q,RA), (SA,RA) (again using the simple SIM_{word} word similarity function) and training with them a J48-graft tree classifier. Table 4.8 shows the results obtained by the soft cardinality system predicting correctness in 5 categories. In all other numbers of categories and evaluation measures, the soft cardinality system also ranked 1^{st} overall datasets [?]. Recently, Leeman-Munk et al. [?] integrated the soft cardinality approach in an experimental automatic tutoring system.

[7]https://translate.google.com

Table 4.7: Results for SemEval task 1 in 2014

	Entailment		**Relatedness**			
system	**accuracy**	**off. rank**	**Pearson**	**Spearman**	**MSE**	**off. rank**
UNAL-NLP_run1	83.05%	3rd/18	0.8043	0.7458	0.3593	4th/17
UNAL-NLP_run2	79.81%	-	0.7482	0.7033	0.4487	-
UNAL-NLP_run3	80.15%	-	0.7747	0.7286	0.4081	-
UNAL-NLP_run4	80.21%	-	0.7662	0.7142	0.4210	-
UNAL-NLP_run5	83.24%	-	0.8070	0.7489	0.3550	-
ECNU_run1	83.64%	2nd/18	**0.8280**	**0.7689**	0.3250	1st/17
Stanford_run5	74.49%	12th/18	0.8272	0.7559	**0.3230**	2nd/17
Illinois-LH_run1	**84.58%**	1st/18	0.7993	0.7538	0.3692	5th/17

Table 4.8: Best results obtained by the soft-cardinality system in the Student Response Analysis task at SemEval 2013 (weighted-average F_1 in 5 correctness levels)

Dataset	**Testing group**	**Size**	**Rank**	**Soft Card.**	**Top Sys.**
Beetle	unseen answers	439	$4^{th}/9$	0.558	0.705
	unseen questions	819	$4^{th}/9$	0.450	0.614
SciEntsBank	unseen answers	540	$4^{th}/9$	0.537	0.625
	unseen questions	733	$1^{st}/9$	0.492	0.492
	unseen domains	4,562	$1^{st}/9$	0.471	0.471
F_1weighted average		**7,093**	$\mathbf{1^{th}/9}$	**0.502**	**0.502**

4.5 Conclusion

We presented our experience participating in SemEval competitions using soft cardinality and cardinality-based feature representations. This article describes the basic methods and particular methods for addressing textual similarity, multilingual textual similarity, typed-textual similarity, textual entailment, cross-lingual textual entailment and automatic students' answer grading. A summary of the official results obtained in SemEval challenges provides the evidence of the effectiveness of the used methods in open competition. It can be come to the conclusion that soft cardinality is a practical and effective tool to address several NLP problems. Furthermore, the soft cardinality model is general enough to be used in other domains and applications.

Chapter 5

Cardinality-based lexical similarity in WordNet:

Bridging the gap to neural embedding

Measuring lexical similarity using WordNet has a long tradition, but has been challenged by distributional methods in the last decade, and more recently by neural word embedding. In the last three years, several larger lexical similarity benchmarks have been proposed where word embedding has achieved state of the art results. The success of such methods has eclipsed the use of WordNet for predicting human judgments of lexical similarity and relatedness. We propose a new method that exploits the WordNet graph, obtaining a word representation based on related neighbor words. Features extracted from cardinalities computed on this word representation, combined with support vector regression and a training dataset based on common synonyms and antonyms, produced competitive results versus word embedding approaches. Moreover, the proposed approach obtained state of the art results in 3 out of the 13 benchmarks tested. Although, word embedding is still the best approach for the task, our method significantly reduced the gap between knowledge based approaches and distributional representations.

A version of this chapter was submitted to *Information Sciences* journal.

5.1 Introduction

WordNet [? ?] is a lexical database that links words in a graph connected by relationships of synonymy, hyperonymy, hyponymy, etc. This has been used for 20 years for addressing many NLP tasks, including lexical similarity. Lexical similarity functions based on WordNet use graph measures to provide a numerical score of the similarity between two *synsets* (senses in WordNet) [? ?]. Another important concept added this approach is *information content* [?], which combines counts of lexical units in corpora aware of the WordNet *is-a* hierarchy [? ? ? ?]. Agirre et al. explored another direction, proposing an adapted version of the PageRank algorithm and extracting a probability distribution over WordNet synsets, which were encoded as vectors for comparison.

The other common approach to address lexical similarity is to make use of the *distributional hypothesis* in the sense of "words with similar meaning will occur with similar neighbors if enough material is available" [?]. This approach involves the construction of a matrix *words* × *contexts*, whose entries contains the number of times a word occurs in a particular context across corpora. The 'context' of a word could be word co-occurrence, a symmetrical sliding window around the word, a sentence, dependencies, etc. The goal of this approach is to obtain a vectorial representation of the words, and combine this with cosine similarity to provide a similarity score. The resultant matrix is large and sparse, which makes it necessary to reduce dimensionality by limiting the size of the word vocabulary or using techniques such as *latent semantic analysis* (LSA) [?], *non-negative matrix factorization* (NMF) [?], and/or *random indexing* [?], among others. Agirre et al. [?] compared some distributional and WordNet approaches, concluding that the former consistently outperformed the latter for lexical similarity. They also differentiated the similarity from the relatedness task, and consequently most benchmarks for lexical semantics clearly differentiate these categories.

Among distributional approaches, neural word embedding [? ? ?] has received great attention. In this approach, rather than first obtaining word contexts then reducing dimensionality, the steps are merged in a single procedure that attempts to learn a language model from corpora with an optimal word representation. This language model aims to build a prediction model for each word given a large number of contexts. Baroni et al. [?] compared traditional distributional methods against word embedding, and concluded that word embedding was superior in performance at several tasks, including lexical similarity and relatedness.

Both WordNet and distributional methods have their pros and cons in practical applications. For example, while WordNet based approaches are aware of the different senses of a word, distributional methods merge senses in a single representation. In contrast, when a language other than English is used, WordNet is a difficult and costly resource to build, but text corpora required by distributional approaches are generally available. Recently, Aletras and Stevenson [?] proposed a hybrid approach combining word embedding and WordNet, obtaining very competitive results but observed only a marginal contribution of the WordNet component to the overall performance.

There is an important gap in performance between WordNet based and word embedding approaches. In this paper we attempt to reduce this difference, and we propose a new method using WordNet to build lexical similarity relatedness functions. The proposed method uses supervised learning and a set based word representation extracted from WordNet neighbor words in the graph. Rather than extracting a predefined set of features from this representation, we extract a relatively large number of features and select an optimal subset in a supervised way. The features are cardinalities (counts) of the two representing sets and of different set operations between them. For training such models, we used a subset of the available benchmarks and a new dataset, W1500, based on a list of common synonyms and antonyms. We demonstrate that models trained with such simple dataset performed similar to models trained with larger datasets labeled with considerably more expensive annotations.

We compared the proposed method against recent studies by Pennington et al. [**?**], Aletras and Stevenson [**?**], and Baroni et al. [**?**] using identical experimental setups. We also use other publicly available benchmarks for testing, making a total of 13 comparisons. We surveyed the state of the art results for these benchmarks and compared them against results obtained in this paper. Our proposed method is a competitive alternative to word embedding for the lexical similarity task, but less competitive for the lexical relatedness task.

5.2 Cardinality-based lexical similarity

The proposed word representation consists of sets containing related words from the neighborhood of a word in the WordNet's graph. The motivation for this is that neighbor words in WordNet are connected by strong semantic relationships that are less noisy than the relationship inferred from corpora statistics. Lexical similarity functions based on WordNet usually have *synsets* as arguments, implying that the words to be compared were extracted from a context and that a previous word sense disambiguation process was performed. The proposed representation is used to model and compare words rather than synsets, allowing a direct comparison against distributional methods. In addition, the proposed method differs from classical WordNet based approaches that focus on a particular relationship (e.g. hypernyms) in that we use all types of relationships available in WordNet.

WordNet is a graph where nodes are synsets connected by directed semantic relationships, e.g. hypernyms, entailments, attributes, etc. Associated with each synset is a textual definition, *gloss*, describing the meaning of the synset. Each synset also has a set of lexical representations, *lemmas*, which in turn can be linked to other lemmas by relationships at lemma level, e.g. antonyms, pertanyms, related forms, etc. Lemmas may also be related indirectly through synsets, thereby inheriting all the synset level relationships. For the proposed representation, the neighbor words (lemmas) of a word (also a lemma) are obtained by following all possible synset and lemma level relationships. The obtained set of words is further enriched by extracting keywords from the textual definitions of the synsets directly related to the word to be represented.

The process, which is detailed in the following subsections, can be summarized as: *i*) take a set of word pairs, each labeled with a gold standard of lexical similarity or relatedness; *ii*) represent each word as a set of related words extracted from WordNet; *iii*) extract 17 cardinality based factors from each pair of representing sets; *iv*) recombine these factors in 272 rational features; *v*) determine a reduced set of features in a supervised way; and *vi*) fit a regression model using the reduced set of features to a gold standard of similarity or relatedness.

5.2.1 Word representation by neighbor words

Let w be a word, then the set of neighbor words related to w, R_w, can be obtained from WordNet using the procedure

1. Let $Synsets_w$ be the set of synsets where the word w belongs to.
2. Let $RelatedSynsets_s$ be the set of synsets related to synset s. This is obtained by the union of the following sets whose names indicate their meaning in WordNet:

$$\begin{aligned} RelatedSynsets_s = \; & Hypernyms_s \cup InstanceHypernyms_s \cup \\ & Hyponyms_s \cup InstanceHyponyms_s \cup \\ & MemberHolonyms_s \cup MemberMeronyms_s \cup \\ & SubstanceHolonyms_s \cup PartHolonyms_s \cup \\ & SubstanceMeronyms_s \cup PartMeronyms_s \cup \\ & Attributes_s \cup Entailments_s \cup Causes_s \cup \\ & AlsoSees_s \cup VerbGroups_s \cup SimilarTo_s \end{aligned}$$

3. The set of synsets related to w is $RelatedSynsets_w = \bigcup_{s \in Synsets_w} RelatedSynsets_s$.
4. Let $Lemmas_s$ be the set of lemmas associated to synset s.
5. The set of lemmas associated to word w is $Lemma_w = \bigcup_{s \in RelatedSynsets_w} Lemmas_s$.
6. Let $RelatedLemmas_l$be the set of lemmas related to lemma l by merging the following sets:

$$\begin{aligned} RelatedLemmas_l = \; & Antonms_l \cup Pertanyms_l \cup TopicDomains_l \cup \\ & RegionDomains_l \cup UsageDomains_l \cup \\ & DerivationallyRelatedForms_l \cup \\ & Hypernyms_l \cup InstanceHypernyms_l \cup \\ & Hyponyms_l \cup InstanceHyponyms_l \cup \\ & MemberHolonyms_l \cup MemberMeronyms_l \cup \\ & SubstanceHolonyms_l \cup PartHolonyms_l \cup \\ & SubstanceMeronyms_l \cup PartMeronyms_l \cup \\ & Attributes_l \cup Entailments_l \cup Causes_l \cup \\ & AlsoSees_l \cup VerbGroups_l \cup SimilarTo_l \end{aligned}$$

Table 5.1: List of additional stopwords from WordNet synsets' definitions

act, act, action, another, become, body, capable, cause, change, coming, consisting, containing, especially, etc, form, giving, group, lacking, made, make, move, one, order, part, particular, people, person, persons, place, position, property, quality, relating, resulting, small, somebody, someone, something, state, time, two, used, using, usually, whose

7. The set of all related lemmas to w is obtained by expanding the set $Lemmas_w$ with their related lemmas:

$$AllRelatedLemmas_w = Lemmas_w \cup \bigcup_{l \in Lemmas_w} RelatedLemmas_l.$$

8. Let $DefinitonKeywords_s$ be the set of words obtained from the textual definition of synset s after removing stopwords[1].The stopwords list was extended with the words showed in Table 5.1, which are frequent and mostly uninformative in WordNet's textual definitions.

9. The set of related words to w is obtained using

$$\mathrm{R}_w = AllRelatedLemmas_w \cup \bigcup_{s \in RelatedSynsets_w} DefinitionKeywords_s.$$

The set, R_w, of related words to w is composed by all its neighbor lemmas and the keywords in the definitions of its related synsets. Table 5.2 shows some examples of representations obtained using the proposed procedure for some words in the rare words (RW) dataset [**?**]. In these examples, the majority of the representing words appear to be meaningfully related to the represented word. Some poorly related words could arise from less common senses of the related synsets or from unrelated words extracted from glosses. Due to space restrictions here, the number of representing words in the selected examples is very low. The average number of representing words in RW is 115, whereas the Stanford Contextual Word Similarities (SCWS) dataset [**?**], which contains less rare words, has an average of 211 representing words for each word. Hence, common words seems to have better connectivity in WordNet's graph compared to less common words.

5.2.2 Features

Once two words, a and b, are represented by their sets of related words, R_a and R_b, respectively, these sets must be compared to provide a similarity score that reflects the degree of similarity or relatedness between them. The first option is to use an off the shelf resemblance coefficient based on cardinality, such as Jaccard [**?**], or Dice [**?**] coefficients. These are rational expressions that combine the three cardinalities, $|\mathrm{R}_a|$, $|\mathrm{R}_b|$ and $|\mathrm{R}_a \cap \mathrm{R}_b|$ (or alternatively $|\mathrm{R}_a \cup \mathrm{R}_b|$),

[1]In our experiments we used the list of English stopwords included in nltk.org

Table 5.2: Examples of the proposed word representation based on neighbor words in WordNet

w	R_w
irrationally	*disorder, insane, insanity, irrational, irrationality, mind, permanent, powers, relatively, understanding, unreason, based, consistent, free, good, healthy, judgment, logic, mental, mentally, normal, powers, rational, rationality, rationalness, reason, sane, saneness, sanity, sense, sound*
subdivide	*apart, carve_up, come, dissever, divide, divider, divisible, division, part, partitive, parts, pieces, portions, separate, smaller, split, split_up, subdivide, subdivider, subdivision, subdivisions belief, common, concert, impossible, indivisible, purpose, undergoing, unite*
wealthy	*abundant, affluence, affluent, flush, loaded, material, money, moneyed, possessing, possessions, rich, richness, supply, value, wealth, wealthiness, wealthy,little, poor, poorness, poverty*

to produce a similarity score in the $[0, 1]$ interval. Most of these coefficients produce metrics with desirable properties such as transitivity. However, generally they fail to adapt to particular tasks, yielding sub-optimal performance. Alternatively, parameterized coefficients [**? ? ? ?**] provide some degree of adaptability, allowing adjustment of parameters using training data.

When training data is available, it would be preferable to use regression for training a similarity function adapted to the particular task. This approach showed to be effective for several NLP tasks in the recent SemEval campaigns [**? ? ?**]. The cardinality based features extracted from a pair of sets comprised cardinalities of all possible areas in their Venn diagram. Some additional features were derived by combining the basic features into rational coefficients in an attempt to capture non-linear relationships. Although some of these feature sets produced quality prediction models, the features sets were somewhat arbitrary and their selection was guided by intuition. For our lexical similarity task, a relatively large set of features was extracted using a set of 17 factors to be combined in rational forms. Table 5.3 shows this set of factors, $f_i, i = 0 \dots 16$, which are combined in rational forms, $\frac{f_i}{f_j}, i \neq j; i, j = 1 \dots n$, to produce a feature. These factors are the building blocks of many resemblance coefficients, e.g. $\frac{f_4}{f_8}$ is the matching coefficient, and $\frac{f_4}{f_{11}}$is the cosine coefficient. Factors f_8 to f_{14} are different instances of the generalized mean between $|\mathrm{R}_a|$ and $|\mathrm{R}_b|$. Factor f_0 allows the factors and their multiplicative inverses to be features. Although the space of possible features is vast, this methodology produces a relatively large feature set of 272 features just from recombining 3 basic cardinalities. The aim of having these features is to use labeled training data for learning not only the similarity function but also a convenient subset of features.

Table 5.3: Factors for rational features

$f_0 = 1$	$f_6 = \|\mathrm{R}_b \setminus \mathrm{R}_a\|$	$f_{12} = \sqrt{\frac{\|\mathrm{R}_a\|^2+\|\mathrm{R}_b\|^2}{2}}$
$f_1 = \|\mathrm{R}_a\|$	$f_7 = \|\mathrm{R}_a \triangle \mathrm{R}_b\|$	$f_{13} = \sqrt[3]{\frac{\|\mathrm{R}_a\|^3+\|\mathrm{R}_b\|^3}{2}}$
$f_2 = \|\mathrm{R}_b\|$	$f_8 = \min(\|\mathrm{R}_a\|, \|\mathrm{R}_b\|)$	$f_{14} = \frac{2\times\|\mathrm{R}_a\|\times\|\mathrm{R}_b\|}{\|\mathrm{R}_a\|+\|\mathrm{R}_b\|}$
$f_3 = \|\mathrm{R}_a \cup \mathrm{R}_b\|$	$f_9 = \max(\|\mathrm{R}_a\|, \|\mathrm{R}_b\|)$	$f_{15} = \min(\|\mathrm{R}_a \setminus \mathrm{R}_b\|, \|\mathrm{R}_b \setminus \mathrm{R}_a\|)$
$f_4 = \|\mathrm{R}_a \cap \mathrm{R}_b\|$	$f_{10} = \frac{\|\mathrm{R}_a\|+\|\mathrm{R}_b\|}{2}$	$f_{16} = \max(\|\mathrm{R}_a \setminus \mathrm{R}_b\|, \|\mathrm{R}_b \setminus \mathrm{R}_a\|)$
$f_5 = \|\mathrm{R}_a \setminus \mathrm{R}_b\|$	$f_{11} = \sqrt{\|\mathrm{R}_a\| \times \|\mathrm{R}_b\|}$	-

5.2.3 Cardinality

In addition to the classical set cardinality, the proposed set of features can also be computed using *soft cardinality* [?], known in the ecology field as *0 order diversity* [?]. Soft cardinality requires an auxiliary similarity measure to compare pairs of elements (words in our scenario) and obtain the soft count of the number of elements in a set. We used the WordNet based measure proposed by Resnik [?], using values of information content computed from the Brown corpus. Since this and most other measures based on WordNet are defined for comparing synsets, we adapted them for comparing pairs of words (lemmas) by averaging the similarity scores of all possible synset pairs in the Cartesian product of the synsets associated to each word:

$$sim(a,b) = \frac{\sum\limits_{(s_a,s_b)\in Synsets_a \times Synsets_b} Resnik(s_a,s_b)}{|Synsets_a \times Synsets_b|} \tag{5.1}$$

Thus, the soft cardinality of a set of words, $\mathrm{W} = \{w_1, w_2, \ldots, w_n\}$, is

$$|\mathrm{W}|' = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} sim(w_i, w_j)^p \right)^{-1} \tag{5.2}$$

Equation 5.2 is used to obtain $|\mathrm{R}_a|'$ and $|\mathrm{R}_b|'$. The soft cardinality of their union is obtained by calculating the soft cardinality of the concatenation of both sets, $|\mathrm{R}_a \cup \mathrm{R}_b|' = |\mathrm{R}_a \oplus \mathrm{R}_b|'$. The intersection is $|\mathrm{R}_a \cap \mathrm{R}_b|' = |\mathrm{R}_a|' + |\mathrm{R}_b|' - |\mathrm{R}_a \cup \mathrm{R}_b|'$, and the remaining factors in Table 5.3 are derived from these 4 basic soft cardinalities. In our experiments, we used $p = 5$ for the softness controller parameter.

5.2.4 Feature selection

The proposed method for feature selection is supervised. Suppose we have a training dataset composed of n pairs of words, where each pair is annotated with a gold standard of similarity or relatedness, and the 272 features described above are extracted for each training pair. This dataset, **D**, is a matrix of size

$n \times 272$, with target vector $\mathbf{T}$, 1×272, containing the gold standard. Let $\mathbf{D}^k$ be a matrix of size $n \times k$ containing a selection with the k best features obtained from a linear regression model that fits $\mathbf{D}$ to $\mathbf{T}$. In our experiments we used the *KBest* feature selection method implemented in *Scikit-learn* [**?**]. To avoid overfitting, the feature selection process was performed on each one of the 10 partitions of a 10 fold cross validation split and the samples were randomly shuffled 30 times for a total of 300 partitions of $\mathbf{D}$ and $\mathbf{T}$. We denote such partitions as $\mathbf{D}_{i,j}, \mathbf{T}_{i,j}$, where i indexes the shuffles and j the folds. The k best features, $D^k_{i,j}$, were obtained for each partition, and those features selected in at least 80% of the 300 partitions were included in the final set of features. Note that altering the 80% inclusion from 65% to 95% did not produce significant changes in the selected feature set.

5.2.5 Regression

Once the features were extracted and selected using either classic or soft cardinality, they were combined using support vector regression [**?**]. The support vector parameters were set to $C = 100$, $\gamma = 0.001$, and the most appropriate kernel for the task was RBF. All features were standardized before training and means and standard deviations were saved to apply the same transformation to the test data. Each regression model was denoted by svr_{DATA}, where $DATA$ indicates the name of the training dataset used.

5.2.6 W1500 an inexpensive training dataset for lexical similarity

Benchmarks for lexical similarity and relatedness were built by selecting a set of pairs of words and aggregating 10 or more human judgments in a fixed numerical scale to obtain a gold standard [**? ? ? ? ?**]. Recent benchmarks, such as MEN [**?**], were built by aggregating 50 binary judgments for each pair in an attempt to reduce the noise and cognitive load of using a numerical scale. This methodology requires a large amount of costly hand labor and limits the convenience of supervised approaches, such as that proposed in this work. There is also the inconvenience that word pairs used in benchmarks are selected with particular criteria (e.g. common nouns), making the models trained with such data less applicable for other type of words.

As an affordable alternative, we built a dataset using list of common synonyms and antonyms taken from publicly available web pages for English as a second language students[2]. We collected 500 pairs of synonyms and 500 pairs of antonyms. The synonym pairs were labeled with a lexical similarity score of 1 and antonym pairs were labeled with a constant c. We experimentally determined $c = 0.2$ to be meaningful for the lexical similarity task. A third subset of 500 pairs was obtained from random combinations of words from the synonyms

[2]`http://www.englishleap.com/vocabulary/synonyms`, `http://www.englisch-hilfen.de/en/words/synonyms.htm`

and antonyms subsets and manually verified that the two words were unrelated. The pairs in this third subset were labeled with a lexical similarity score of 0.

We refer this dataset as W1500[3]. The aim of this dataset was to show that supervised models for predicting lexical similarity, trained with such simple resource, can perform similarly to models trained with current benchmarks.

5.3 Experiments

The experiments compare the performance of the proposed lexical similarity relatedness measure with classical measures based solely on WordNet and with scores obtained from cosine similarities between word representations obtained from word embedding. Our main interest is to determine which data is more convenient for training our models and to what extent our approach is a competitive alternative to word embedding approaches.

5.3.1 Datasets

We used the benchmarks employed by Pennington et al. (2014) [?] for word similarity: WS353: Finkelstein et al., 2001 [?], MC: Miller and Charles, 1991 [?], RG: Rubenstein and Goodenough, 1965 [?], SCWS: Huang et al., 2012 [?], and RW: Luong et al., 2013 [?]. Since the proposed method is supervised, we combined these 5 datasets in all possible combinations of training/testing and compared results using the proposed W1500 dataset for training. Our method was compared using the same experimental setup employed by Pennington et al.

We also compared our proposed approach with Aletras and Stevenson (2015) [?]. The comparison used the same 6 datasets as they employed: MC, RG, WS353, the semantic (WSS) and relatedness (WSR) partitions of WS353 proposed by Agirre et al. [?], and the MEN dataset (relatedness) proposed by Bruni et al. [?]. For the sake of future comparison, we provide a complete combination of all possible configurations of training/testing using the 8 standard benchmarks and also our W1500 dataset.

For additional testing of our approach, we perfomed further experiments using datasets: YP-130 (similarity) [?], MTURK287 (relatedness) [?], MTURK771 (relatedness) [?], Rel-122 (relatedness) [?], Verb-143 [?], and the recently proposed SimLex-999 (similarity) [?]. All these datasets were not observed during the training and development of the method proposed in this paper.

Summarizing the results of Pennington et. al [?], Aletras and Stevenson [?], the recent survey of Baroni et al. [?], and the results of the current work (and others), we present an updated survey of the state of the art for the 13 benchmarks employed and W1500.

[3]`https://sites.google.com/site/sergiojimenezvargas/W1500.txt`

5.3.2 Performance measures

The customary measure for assessing the performance of a lexical similarity method is the Spearman's correlation coefficient, ρ. To provide a single measure across different sets of datasets we propose the simple average, $\bar{\rho}$, of ρ obtained on each dataset. We consider the average weighted by the number of samples in each dataset, similar to the *mean* measure used for aggregating Pearson's coefficients across different datasets in the semantic textual similarity task in the recent SemEval campaigns [?]. However, we opted for the simple average as the number of samples in the benchmarks varies considerably from 30 in MC to 3,000 in MEN. The aim of this measure is to compare the overall performance of each method against results published by Pennington et al. [?], Aletras and Stevenson [?], and Baroni et al. [?]. Thus, $\bar{\rho}_{Penn}$ are averages across MC, RG, WS353, SCWS, and RW datasets; $\bar{\rho}_{Ale}$ are averages across MC, RG, WS353, WSS, WSR, and MEN; and $\bar{\rho}_{Bar}$ are averages across RG, WS353, WSS, WSR, and MEN. For consistency with published results, $\bar{\rho}_{Ale}$ and $\bar{\rho}_{Bar}$ are reported with 2 decimals, and $\bar{\rho}_{Penn}$ with 4. Note that none of the $\bar{\rho}$ measures include W1500, our proposed training dataset.

5.3.3 Baselines

The first baseline for the lexical similarity task is a morphological measure based on the *edit distance* [?]. This baseline is

$$sim(a,b) = 1 - \frac{ed(a,b)}{max(len(a).len(b))},$$

where $ed(a,b)$ measures the minimum number of edit operations (i.e., insertion, deletion, and replacement) needed at character level to convert word a into b or vice versa.

The second group of baselines comprises classical lexical similarity measures based on WordNet implemented in the NLTK [?]: *path*: the inverse of the number of edges between to synsets, *lch*: Leacock and Chodorow[?], *wup*: Wu and Palmer [?], *lin.brown*/*lin.semcor*: Lin's measure using Brown or Semcor corpora for information content calculation, *res.brown*/*res.semcor*: Resnik [?], and *jcn.brown*/*jcn.semcor*: Jiang and Conrath[?].

Table 5.4 shows the performance measures across all datasets and the number of word pairs on each one (final row). The edit distance baseline is rather low in all measures, and only SCWS and RW datasets include morphologically similar word pairs. The classical measures based on WordNet are particularly unsuitable for modeling word relatedness and infrequent words given the poor performances in WSR (relatedness), MEN (relatedness), and RW (rare words) datasets. This inability for modeling relatedness by classic WordNet measures was also observed by Szumlanki et al. [?] using another dataset labeled with relatedness judgments (Rel-122). All baselines obtained $\bar{\rho} < 0.47$.

Table 5.4: Spearman's rank correlation of edit distance and classic similarity functions based on WordNet for lexical similarity and relatedness benchmarks

dataset $\rightarrow$ *measure* $\downarrow$	**MC**	**RG**	**WS353**	**WSS**	**WSR**	**MEN**	**SCWS**	**RW**	**W1500**	$\bar{\rho}_{Penn}$	$\bar{\rho}_{Ale}$	$\bar{\rho}_{Bar}$
edit dist.	-0.1642	0.0690	-0.0181	0.0063	-0.0487	0.0771	0.2851	**0.1818**	-0.0145	0.0707	-0.01	0.02
path	0.6035	0.6610	0.2966	0.5483	0.0250	0.2717	0.4435	0.0497	0.2545	0.4109	0.40	0.36
lch	0.5176	0.6410	0.2387	0.4607	-0.0088	0.2589	0.3949	0.0146	0.0855	0.3613	0.35	0.32
wup	0.5425	0.4567	0.3232	0.5359	0.0856	0.2160	0.3694	0.0141	0.1996	0.3412	0.36	0.32
lin.brown	0.7144	0.6223	0.3319	0.6105	0.0161	0.2713	0.3697	0.0121	0.1499	0.4101	0.43	0.37
lin.semcor	0.6647	0.4868	0.2238	0.4639	-0.0204	0.2619	0.3943	0.0283	0.1567	0.3596	0.35	0.28
res.brown	0.6742	0.6464	0.3630	**0.6218**	0.0460	0.2743	0.3822	0.0582	0.1479	0.4248	0.44	0.39
res.semcor	0.7037	0.6563	**0.3664**	0.6191	**0.0587**	0.2810	0.4317	0.0531	0.1570	0.4422	0.45	**0.40**
jcn.brown	**0.8197**	**0.7272**	0.3099	0.6120	-0.0047	**0.2928**	**0.4556**	0.0122	**0.3263**	**0.4649**	**0.46**	0.39
jcn.semcor	0.5934	0.4108	0.1587	0.3497	-0.0574	0.1966	0.4203	0.0049	0.3246	0.3176	0.28	0.21
# pairs	30	65	353	203	252	3000	1997	2034	1500	-	-	-

5.3.4 Number of features

Given that the feature selection method proposed in subsection 5.2.4 is supervised, the use of 10 fold cross validation is important to determine the k-best parameters for each training dataset, avoiding overfitting. As k ranged from 5 to 200 features, the Spearman's coefficient, ρ_{CV}, was computed for each dataset and each cardinality function (i.e., classic and soft), comparing cross validated predictions versus gold standards. The configuration with the best ρ_{CV} for pair dataset cardinality pair and smallest number of k-best features was selected. This criterion aims to select a k-best value that yields a near optimal performance while keeping the number of features small. The k-best values for those selected configurations are reported in Table 5.5 (columns ρ_{CV}).

Although we expected good correspondences between the ρ_{CV} columns, such as those obtained by svr_{MC} and svr_{WSR}, most of the trained models show significant differences of the optimal k-best between classic and soft cardinalities. For example, model svr_{WS353} obtained an unexpectedly high number of features (128 features for classic cardinality) for a dataset with only 353 samples. Consequently, we determined the optimal k-best values for the best configurations of $\bar{\rho}_{Penn}$ and $\bar{\rho}_{Ale}$ test measures. The k-best values determined at training time for soft cardinality are good predictors of their corresponding optimal test values, but differs significantly in most cases for classic cardinality. Nevertheless, in the following experiments the k-best values employed for each dataset cardinality pair are those in columns ρ_{CV} in Table 5.5 (values in bold face).

Table 5.5: Optimal values of the k-best parameter for feature selection method

		Classic cardinality			Soft cardinality		
regressor	training samples	ρ_{CV}	$\bar{\rho}_{Penn}$	$\bar{\rho}_{Ale}$	ρ_{CV}	$\bar{\rho}_{Penn}$	$\bar{\rho}_{Ale}$
svr_{MC}	30	**17**	17	17	**21**	21	21
svr_{RG}	65	**8**	25	15	**26**	32	32
svr_{WS353}	353	**128**	12	16	**45**	45	45
svr_{WSS}	203	**16**	25	15	**52**	27	52
svr_{WSR}	252	**14**	29	29	**14**	14	14
svr_{SCSW}	1,997	**74**	27	15	**65**	61	61
svr_{RW}	2,034	**128**	9	9	**32**	30	30
svr_{MEN}	3,000	**87**	8	10	**79**	65	65
svr_{W1500}	1,500	**91**	35	32	**76**	51	51

5.3.5 Previous results from distributional and word embedding methods

Table 5.6 shows the Spearman's correlations of recently published studies addressing lexical similarity relatedness tasks using word embedding methods. The first group corresponds to the work of Pennington et al. (2014) [**?**] who proposed GloVe, a word embedding method that combines evidence from local context and global counts. GloVe vectors are compared against word2vec (CBOW) [**?**] and a SVD baseline, all of them in a 300 dimension space and trained on the same corpora. The name of each method includes the size of the training corpora in billions of tokens. The best performing method in this group is 'GloVe 42B', which produces $\bar{\rho}_{Penn} = 0.6996$.

The second group in Table 5.6 corresponds to Aletras and Stevenson (2015) [**?**]. They proposed several hybrid models that combined word embedding and WordNet (H models) and compared them with a word embedding trained with a 2.8B token corpora (D model). H^* is the best method of this group, with $\bar{\rho}_{Ale} = 0.72$ and $\bar{\rho}_{Bar} = 0.71$.

The final group of results are taken from a survey by Baroni et al. (2014), where they compared word2vec embedding ('predict' models) against models based on token counts in corpora, i.e., distributional semantic models (DSM), combined with dimensionality reduction techniques, such as SVD or NMF. They proposed 48 configurations of the former and 36 of the latter, varying parameters such as final dimensionality, context window size, and others. Results 'each' correspond to the best configuration for each benchmark and 'all' to the single best configuration across all benchmarks. Models in this group were trained using the same 2.8B token corpora used by Aletras and Stevenson. Methods D and 'count.each' are equivalent and, as expected, obtain very similar results ($\bar{\rho}_{Bar}$).

The best performing method in groups 2 and 3 is 'predict.each', $\bar{\rho}_{Bar} = 0.78$. Model H^* was outperformed by 'predict.all', which are a more comparable pair since both have a single configuration across all benchmarks. Unfortunately,

Table 5.6: Spearman's rank correlation from state of the art methods published by Pennington et al., Aletras & Stevenson, and Baroni et al. ([? ? ?])

method	**MC**	**RG**	**WS353**	**WSS**	**WSR**	**MEN**	**SCWS**	**RW**	$\bar{\rho}_{Penn}$	$\bar{\rho}_{Ale}$	$\bar{\rho}_{Bar}$
SVD-L 6B	0.7270	0.751	0.6570	na	na	na	0.5650	0.3700	0.6140	na	na
SVD-L 42B	0.7640	0.741	0.7400	na	na	na	0.5830	0.399	0.6454	na	na
CBOW 6B	0.6560	0.6820	0.5720	na	na	na	0.5700	0.3250	0.5610	na	na
CBOW 100B	0.7960	0.7540	0.6840	na	na	na	0.5940	0.4550	0.6566	na	na
GloVe 6B	0.7270	0.7780	0.6580	na	na	na	0.5390	0.3810	0.6166	na	na
GloVe 42B	**0.8360**	**0.8290**	**0.7590**	na	na	na	**0.5960**	**0.4780**	**0.6996**	na	na
D	0.72	0.79	0.62	0.70	0.59	**0.72**	na	na	na	0.69	0.68
H	0.64	0.69	0.49	0.60	0.36	0.58	na	na	na	0.56	0.54
H_p	0.86	0.82	0.58	0.67	0.49	0.63	na	na	na	0.68	0.64
$\tilde{H}$	0.71	0.71	0.55	0.69	0.42	0.54	na	na	na	0.60	0.58
$\tilde{H}_p$	**0.86**	**0.85**	0.58	0.68	0.46	0.55	na	na	na	0.66	0.62
H^*	0.79	0.81	**0.67**	**0.76**	**0.57**	**0.72**	na	na	na	**0.72**	**0.71**
H_p^*	0.80	0.86	0.52	0.62	0.41	0.56	na	na	na	0.63	0.59
count.each	na	0.74	0.62	0.70	0.59	0.72	na	na	na	na	0.67
predict.each	na	**0.84**	**0.75**	**0.80**	**0.70**	**0.80**	na	na	na	na	**0.78**
count.all	na	0.70	0.62	0.70	0.57	0.72	na	na	na	na	0.66
predict.all	na	0.83	0.73	0.78	0.68	**0.80**	na	na	na	na	0.76

the results of Pennington et al. are not comparable with those of Aletras and Stevenson, and Baroni et al., but the results presented in the following subsection are comparable with all of them.

Note that there is significant difference between the result obtained by classic WordNet based measures (Table 5.4) and other methods (Table 5.6), especially for WS353, WSR, MEN, and RW datasets. The WordNet based measures obtained competitive results only for the three smallest datasets (MC, RG, and WSS), which contains very common word pairs linked by similarity rather than relatedness.

5.3.6 Results of our proposed method

We present results obtained from the proposed method as presented in section 5.2, using either classic or soft cardinality (Table 5.7 and 5.8, respectively). Rows labeled "*cosine*" shows the results obtained by another baseline that uses the cardinalities $|\mathrm{R}_a|$, $|\mathrm{R}_b|$, and $|\mathrm{R}_a \cap \mathrm{R}_b|$ and combines them with the cosine coefficient,

$$cosine(\mathrm{R}_a, \mathrm{R}_b) = \frac{|\mathrm{R}_a \cap \mathrm{R}_b|}{\sqrt{|\mathrm{R}_a| \cdot |\mathrm{R}_b|}}.$$

Although this is a set based formulation, it is equivalent to the well-known geometrical formulation of the cosine similarity, where R_a and R_b are represented as binary vectors in a vector space model indexed by vocabulary. This baseline allows us to quantify the benefit of using the proposed supervised regression function rather than a plausible off the shelf cardinality based resemblance coefficient.

Each row shows the Spearman's coefficients obtained by our regression function trained with a particular dataset. The best results for each dataset and measure ($\bar{\rho}$) are shown in bold face. Since all possible configurations of training and test data are reported, the diagonal in the results tables shows the training error measured by the mean correlations obtained using 30 rounds of 10 fold cross validation (underlined values). The diagonal data are copied in the row labeled "svr_{self}", along with their corresponding standard deviations, σ, and $\bar{\rho}$.

5.3.6.1 Cosine baseline

Regarding the cosine baseline, there are significant differences in performance from our proposed method combining the set based word representation with a supervised function versus a simple resemblance coefficient. Even our worst performing models doubled the performance baseline. Thus, the complexity of our proposed approach resides in the regression function rather than in the word representation. In contrast, word embedding vectors are usually combined with cosine similarity, which is a conceptually simple function, implying that the complexity of these models lies in the word representation.

5.3.6.2 Classic vs. soft cardinality

Comparing cosine baselines from Tables 5.7 and 5.8, classic cardinality outperforms soft cardinality. In contrast, the proposed regression function suggests soft cardinality to be the better choice. Almost 84% of the entries in Table 5.8 are superior to their counterparts in Table 5.7. Combining this outcome with the convenience of using soft rather than classic cardinality in feature selection (Section 5.2.4) suggests soft cardinality is more appropriate for the task.

5.3.6.3 $\bar{\rho}_*$ measures

Our proposed approach (Tables 5.7 and 5.8) is comparable with word embedding systems (Table 5.6). The best result, $\bar{\rho}_{Penn} = 0.6996$, obtained by GloVe 42B is very close to $\bar{\rho}_{Penn} = 0.6814$ obtained by our model svr_{SCWS} using soft cardinality. However, the SCWS dataset was used as training data for svr_{SCWS}, thus including the training data in part for results measured by $\bar{\rho}_{Penn}$. A better comparison is $\bar{\rho}_{Penn} = 0.6672$ obtained by svr_{W1500} using soft cardinality, which used training data not involved in the comparison measures $\bar{\rho}_*$. Although we do not provide a measure of the significance of the differences in $\bar{\rho}_{Penn}$, the results obtained by svr_{W1500} can be considered competitive given the affordability of the training data (Section 5.2.6) and simplicity of the word representation (Section 5.2.1). Comparing across the datasets of $\bar{\rho}_{Penn}$, any of our svr_* models are comparable aside from WS353, which shows significant differences (0.7590 by GloVe 42B, 0.6102 by our best model).

We can compare our WordNet based model with hybrid WordNet word embedding approaches using $\bar{\rho}_{Ale}$. The baseline of this group is the D model (word embedding only), with $\bar{\rho}_{Ale} = 0.69$, which is equaled by our models svr_{SCWS} and svr_{W1500}, $\bar{\rho}_{Ale} = 0.70$ and 0.69, respectively, using soft cardinality. The best hybrid model, $H*$, produces $\bar{\rho}_{Ale} = 0.72$, and this difference can be attributed to the contribution of WordNet to that model. Across datasets, our soft cardinality models have superior performance for MC; comparable for RG, WS353, and WSS; and inferior for WSR and MEN. Given that WSS (similarity) and WSR (relatedness) are partitions of WS356, this suggests that the proposed method is more suitable for modeling similarity than relatedness, in comparison with word embedding methods that appear to address both facets of lexical analysis properly.

Comparing with Baroni et al., measured by $\bar{\rho}_{Bar}$, shows that word embedding can be refined to fit the lexical similarity relatedness task. Compared with their results, our model is equivalent for 'count' models, but significantly superior for 'predict' (word embedding) models ($\bar{\rho}_{Bar} = 0.78$) obtained by 'predict.each' against 0.66 and 0.65 by our best models.

5.3.6.4 Training performance

The training performance of each model (row svr_{self} in Tables 5.7 and 5.8) appears a good predictor of the performance of other svr_* models trained with different data and tested with the same dataset (values in the same column). As expected, this value is the best performance achieved for large datasets, such as MEN, SCWS, RW, and W1500. The differences between these training performances and models trained with another data are not significant. In particular, using the W1500 dataset for training produced very competitive results in comparison with the training performance (compare svr_{W1500} and svr_{self} rows in tables 5.7 and 5.8). Standard deviations in training may be considered small for all datasets ($\sigma < 0.002$ for large and $\sigma < 0.03$ for small datasets). This shows the stability of the regression model across folds and training test splits and the ability of the proposed method to predict unseen data.

5.3.6.5 Additional test data

Tables 5.9 and 5.10 shows the 9 trained models from svr_{MC} to svr_{W1500} tested in YP-130, MTurk287, MtTurk771, Rel-122, Verb-143, and SimLex-999 datasets. Tables 5.9 and 5.10 show the features extracted using classic and soft cardinality, respectively.

Contrary to the results obtained using training and development data, here classic cardinality consistently outperforms soft cardinality. Classic cardinality yields state of the art results for YP-130 and SimLex-999 datasets (Table 5.11). In contrast, results obtained using Mturk287, Mturk711, and Rel-122, which are datasets containing mainly relatedness pairs, were low. This confirms our previous observation that the proposed method is more suitable for modeling

Table 5.7: Spearman's correlation of the proposed similarity measure for different training/test configurations using classic cardinality. When training and test datasets are the same (underlined diagonal), 10 fold cross validation was used and the result is the average of 30 × 10 fold cross validation rounds.

method	**MC**	**RG**	**WS353**	**WSS**	**WSR**	**MEN**	**SCWS**	**RW**	**W1500**	$\bar{\rho}_{Penn}$	$\bar{\rho}_{Ale}$	$\bar{\rho}_{Bar}$
cosine	0.4205	0.4116	0.2379	0.3880	0.0983	0.2444	0.2330	0.0534	0.3522	0.2713	0.30	0.28
svr_{MC}	$\underline{0.8148}$	0.7856	0.5634	**0.7489**	0.3558	0.6005	0.5950	**0.4477**	0.8035	0.6413	0.64	0.61
svr_{RG}	0.8744	$\underline{0.7592}$	0.5223	0.7344	0.3123	0.6082	0.5533	0.3611	0.7988	0.6141	0.64	0.59
svr_{WS353}	0.7568	0.6878	$\underline{0.5483}$	0.7214	0.3735	0.5487	0.5724	0.2793	0.7555	0.5689	0.61	0.58
svr_{WSS}	**0.8970**	**0.7983**	0.5441	$\underline{0.7436}$	0.3313	0.6227	0.5648	0.4117	0.8020	0.6432	**0.66**	0.61
svr_{WSR}	0.6026	0.6369	0.5161	0.6945	$\underline{0.3123}$	0.6174	0.2723	0.2108	0.6989	0.4477	0.56	0.56
svr_{MEN}	0.8244	0.7632	0.5746	0.7271	0.3775	$\underline{\mathbf{0.6369}}$	0.5609	0.2641	0.7754	0.5974	0.65	0.62
svr_{SCWS}	0.8371	0.7725	**0.5948**	0.7419	**0.3822**	0.6008	$\underline{\mathbf{0.6082}}$	0.4353	0.7984	**0.6496**	0.65	**0.62**
svr_{RW}	0.7374	0.6811	0.5243	0.6817	0.2840	0.4901	0.5714	$\underline{0.4476}$	0.7525	0.5924	0.57	0.53
svr_{W1500}	0.8507	0.7273	0.5688	0.7233	0.3647	0.6140	0.5703	0.4175	$\underline{\mathbf{0.8058}}$	0.6269	0.64	0.60
svr_{self}	$\underline{0.8148}$	$\underline{0.7592}$	$\underline{0.5483}$	$\underline{0.7436}$	$\underline{0.3123}$	$\underline{0.6369}$	$\underline{0.6082}$	$\underline{0.4476}$	$\underline{0.8058}$	0.6356	0.64	0.60
σ	0.0276	0.0142	0.0064	0.0036	0.0108	0.0014	0.0017	0.0014	0.0010	-	-	-

Table 5.8: Spearman's correlation of the proposed similarity measure for different training/test configurations using SOFT cardinality. When training and test datasets are the same, 10 fold cross validation was used and the result is the average of 30 × 10 fold cross validation rounds.

method	**MC**	**RG**	**WS353**	**WSS**	**WSR**	**MEN**	**SCWS**	**RW**	**W1500**	$\bar{\rho}_{Penn}$	$\bar{\rho}_{Ale}$	$\bar{\rho}_{Bar}$
cosine	0.4044	0.4373	0.0996	0.2114	0.0093	0.1957	0.2386	-0.0032	0.2944	0.2354	0.23	0.19
svr_{MC}	$\underline{0.8479}$	0.8358	0.5482	0.7551	0.3315	0.5974	0.5912	0.4431	0.7978	0.6532	0.65	0.61
svr_{RG}	0.8952	$\underline{0.8170}$	0.5333	0.7447	0.3213	0.5979	0.5739	0.4299	0.7961	0.6499	0.65	0.60
svr_{WS353}	0.8769	0.8364	$\underline{0.5650}$	**0.7744**	**0.4396**	0.6362	0.6130	0.4239	0.8030	0.6630	0.69	0.65
svr_{WSS}	**0.9068**	0.8156	0.5961	$\underline{0.7494}$	0.4035	0.6086	0.6064	0.4151	0.7956	0.6680	0.68	0.63
svr_{WSR}	0.8133	0.7683	0.5285	0.7313	$\underline{0.2967}$	0.6032	0.3534	0.3298	0.7600	0.5587	0.62	0.59
svr_{MEN}	0.8756	0.8275	0.5994	0.7561	0.4206	$\underline{\mathbf{0.6435}}$	0.5929	0.3715	0.7932	0.6534	0.69	0.65
svr_{SCWS}	0.8992	0.8417	**0.6102**	0.7721	0.4240	0.6300	$\underline{\mathbf{0.6184}}$	0.4376	0.8036	**0.6814**	**0.70**	**0.66**
svr_{RW}	0.8823	**0.8424**	0.5858	0.7667	0.3934	0.6402	0.6121	$\underline{\mathbf{0.4453}}$	0.8055	0.6736	0.69	0.65
svr_{W1500}	0.8856	0.8203	0.6015	0.7586	0.4178	0.6411	0.5934	0.4352	$\underline{\mathbf{0.8145}}$	0.6672	0.69	0.65
svr_{self}	$\underline{0.8479}$	$\underline{0.8170}$	$\underline{0.5650}$	$\underline{0.7494}$	$\underline{0.2967}$	$\underline{0.6435}$	$\underline{0.6184}$	$\underline{0.4453}$	$\underline{0.8145}$	0.6587	0.65	0.61
σ	0.0276	0.0120	0.0100	0.0064	0.0100	0.0012	0.0010	0.0010	0.0006	-	-	-

Table 5.9: Spearman's correlation coefficients for trained models with classic cardinality features for additional test datasets

model	**YP-130**	**MTurk287**	**MTurk771**	**Rel-122**	**SimLex-999**	**Verb-143**
svr_{MC}	0.7546	0.4223	**0.5498**	0.4820	0.5047	**0.3633§**
svr_{RG}	0.7439	0.5005	0.5178	0.4789	0.5008	0.3503
svr_{WS353}	0.6843	0.3869	0.5214	0.2439	0.4994	0.2396
svr_{WSS}	0.7557	**0.5099**	0.5437	**0.4897**	0.5108	0.3577
svr_{WSR}	0.7185	0.5004	0.4612	0.4819	0.4456	0.2521
svr_{MEN}	0.7492	0.4991	0.5307	0.4586	0.5149	0.3292
svr_{SCWS}	**0.7574†**	0.4816	0.5416	0.3275	0.5215	0.3174
svr_{RW}	0.7121	0.3354	0.5389	0.0940	0.4794	0.3212
svr_{W1500}	0.7385	0.4524	0.5447	0.4711	**0.5327**	0.3280

†Pearson's correlation $r = 0.7970$
§Pearson's correlation $r = 0.6134$

Table 5.10: Spearman's correlation coefficients for trained models with soft cardinality features for additional test datasets

model	**YP-130**	**MTurk287**	**MTurk771**	**Rel-122**	**SimLex-999**	**Verb-143**
svr_{MC}	0.6604	0.3978	0.4862	0.4046	0.3532	0.2724
svr_{RG}	**0.7058**	0.4465	0.4713	0.4071	0.3522	0.2598
svr_{WS353}	0.6088	0.4382	0.4532	0.4020	0.3930	-0.0125
svr_{WSS}	0.5880	0.4048	0.4593	0.4132	0.3782	0.0466
svr_{WSR}	0.6817	0.4669	**0.5074**	0.4117	0.3640	0.2785
svr_{MEN}	0.5691	0.4632	0.4359	**0.4533**	0.3815	-0.0827
svr_{SCWS}	0.5743	0.4224	0.4291	0.3670	0.3875	-0.028
svr_{RW}	0.6558	**0.4942**	0.4725	0.4258	0.3957	-0.0258
svr_{W1500}	0.5595	0.4350	0.4508	0.3629	**0.4034**	-0.0149

lexical similarity than relatedness. It is also important to highlight that the two best models, svr_{WSS} and svr_{W1500}, were trained with a set of only 203 similar word pairs (WSS) and a simple set of common synonyms and antonyms (W1500), respectively.

5.3.7 Updated state of the art on lexical similarity

Baroni et al. [**?**] compiled state of the art results for several benchmarks and tasks including lexical similarity. We extend this by including datasets MC, SCWS, and RW, and update it with the results of Pennington et al. [**?**], this work, and others.

In the last two columns of Table 5.11, we show a comparison of our method against the state of the art results for each benchmark regardless of the methods and resources used. Column 'best' shows the best results obtained for each dataset using any of the proposed models trained with any data, and column 'diff.' shows the relative difference between this and the best result published to date. The larger differences are those of the benchmarks that include, partially or totally, word pairs associated by relatedness, i.e., WS353, WSR, MEN, MTurk287, MTurk711, and Rel-122. These differences range from 37.2% (WSR) to 8.3% (Rel-122), which are significant. In the other benchmarks, characterized by similarity relationships in their word pairs, our proposed method obtained either state of the art results or differences less than 6.4%.

5.4 Discussion

There is a significant gap between the performance of classic WordNet measures and distributional methods, and the gap is larger for neural embedding. The main weaknesses of these measures are their inability to model relatedness and address rare words. This is clearly revealed by the WSR (relatedness part of WS353) and RW (rare words) datasets, where the best scores are extremely low ($\rho = 0.0587$ and $\rho = 0.0582$, respectively). The former is possibly due to the dominant structure in WordNet being the *is a* hierarchy, while the latter could be caused by the few relationships modeled for infrequent words in WordNet. Although our proposed measure does not stand out in these and other similar datasets (Rel-122, MTurk287, and MTurk771), we believe that our approach remedies these drawbacks. A possible explanation is that the proposed measure exploits all types of relationships in WordNet and keywords extracted from synset definitions.

Supervised learning is generally criticized for its dependency on costly labeled data, which also needs to be a representative sample of the target data. However, the proposed approach was effective with a diversity of training data. For example, in Table 5.7, svr_* models trained with any dataset (except WSR) obtained $\bar{\rho}_{Penn} > 0.6499$, comparable to the best result obtained by Pennington et al. ($\bar{\rho}_{Penn} = 0.6996$). In the same table, robustness regarding training data is also evident for models trained using WS353 and its partitions WSS and WSR. While the difference in $\bar{\rho}_{Penn}$ between WSS and WSR is considerable (0.6680 vs. 0.5587), the difference between WS353 (0.6630) and WSS is small despite WS353 containing all WSR data. Similarly, svr_{MC}, which has only 30 word pairs for training, achieved $\rho = 0.5912$ predicting similarities in the SCWS dataset with 1,997 word pairs. This is comparable to the state of the art result for this dataset, $\rho = 0.6184$ (comparing words out of context).

Our proposed model achieves almost the same results when trained with W1500 or SCWS, which was the best training dataset. Thus, the similarity function learned by support vector regression can also be obtained by constraints imposed by synonyms, antonyms, and random pairs of unrelated words. This approach is comparable to that of Halawi et al. [**?**], who used similar constraints to improve a distributional word representation.

Unfortunately our results do not provide a clear conclusion to the use of classic or soft cardinality for computing counts in word representation. On the one hand, soft cardinality seems a clear choice for the 9 datasets used for train-

Table 5.11: Updated state of the art (soa) for the lexical similarity task

dataset	**type**	**soa'sρ**	**method**	**reference**	**best**	**diff.**
MC	sim.	0.91	WordNet context vectors	Patwardhan and Pedersen [?]	0.9068	0.4%
RG	sim.	0.90	WordNet context vectors	Patwardhan and Pedersen [?]	0.8424	6.4%
WS353	both	0.81	constrained word embeddings	Halawi et al. (2012) [?]	0.6102	24.7%
WSS	sim.	0.80	word embeddings	Baroni et al. (2014) [?]	0.7744	3.2%
WSR	rel.	0.72	word embeddings	Agirre et al. (2009) [?]	0.4396	37.2%
SCWS†	sim.	0.6184	this paper	before $\rho = 0.6104$ Li et al. (2014) [?]	0.6184	0.0%
RW	sim.	0.4780	word embeddings	Pennington et al. (2014) [?]	0.4477	6.3%
MEN	rel.	0.80	word embeddings	Baroni et al. (2014) [?]	0.6435	19.6%
YP-130	sim.	0.7574	this paper	before $r = 0.747$ Taieb et al. [?]	0.7574	0.0%
		$r = 0.7970$		or $\rho = 0.71$ [?] (using only 80 pairs)		
MTurk287	rel.	0.737	constrained word embeddings	Halawi et al. (2012) [?]	0.5099	30.8%
MTurk771	rel.	0.727	constrained word embeddings	Halawi et al. (2012) [?]	0.5498	24.4%
Rel-122	rel.	0.534	Lexical co-occurrence in Wikip.	Szumlanski et al. [?]	0.4897	8.3%
SimLex-999	sim.	0.5327	this paper	before $\rho = 0.52$ Hill et al. [?]	0.5327	0.0%
Verb-143	sim.	$r = 0.642$	aggregation of SCF vectors	Baker et al. [?]	$r = 0.6134$	4.4%
W1500	sim.	0.8145	this paper	proposed in this paper.	0.8145	0.0%

†compared only with approaches that do not make use of context sentences or POS tags in SCWS

ing and testing (MC, RG, etc.). For those datasets, soft cardinality performed consistently better than classic, produced less variance in performance across different training data, and the optimal number of selected features in training data was a good prediction for test data. On the other hand, in the experiments on test datasets (YP-130, MTurk*, etc.), classic cardinality consistently outperformed soft cardinality. This behavior could be due to the sensitivity of soft cardinality to its parameters, i.e., the softness control (p) and the auxiliary similarity function (sim) of Eq.5.2. The selection of $p = 5$ and $sim = res.semcor$ was based on results from previous work and a small exploration using RG and WS353 as development datasets. Further exploration of these parameters using more data and cross validation could provide better insight to their effects on performance.

5.5 Conclusion

We presented a supervised learning approach to build a lexical similarity function based on WordNet. This approach was tested in all lexical similarity relatedness benchmarks proposed to date, obtaining state of the art or comparable results when used to predict human judgments of similarity. The proposed method showed to be less effective regarding relatedness, but still superior to other classic measures based on WordNet. Similarly, our method produced comparable results to word embedding approaches for the lexical similarity task, but significant differences were evident when modeling lexical relatedness.

Chapter 6

Conclusion

6.1 Conclusions

The main observation made in this dissertation is that set-based methods have been underestimated in automatic text analysis and processing applications. We have shown that set-based approaches can compete and collaborate with vector, probability, and stringology -based approaches. This is possible in part because of the two main contributions of this work, namely soft cardinality and cardinality-based representations. On the one hand, soft cardinality is a new artifact that was thoroughly studied and developed in this dissertation, making it a useful tool with the potential to be applied in many domains. Our theoretical presentation of the soft cardinality and its connection with the Leinster and Cobbold's zero-order diversity theory provides a solid framework for investigation of new theories and applications. It is now clear that soft cardinality is not only a method for making "soft" counts of the number of elements in collections, but also a measure of the amount of information (in the context of information theory) and diversity (in the context of individuals, species, and communities.)

On the other hand, Cardinality-based representations showed its potential for addressing various practical problems in text applications. From now on, there is a new alternative aside from the resemblance coefficients for using classic and soft cardinality to build similarity functions, classifiers and others. Resemblance coefficients are the fundamental components of the representations based on cardinality. However, the combination of these representations with machine learning techniques provides a powerful tool for modeling the complex interactions between objects. In this dissertation, only some sets of feature were explored, opening the way for new questions to be formulated and investigated.

6.2 The takeaways

We are excited about the possible applications of soft cardinality in the near future. Besides suggested NLP and computer science applications, we expect to witness applications such as measuring the diversity of students in a university or comparing political regions for social or economic studies. Consider measuring the demographic diversity of your department at your university by representing it as the set of its students. Simply provide a pairwise similarity measure between students, which can be obtained by representing each student as a set of demographic features and compare pairs using Jaccard's index. Let D be a department and s a student:

$$\mathrm{div}(D) = \sum_{s_a \in D} \frac{1}{\sum_{s_b \in D} sim(s_a, s_b)}$$

Now, let's find out if academic performance is correlated with diversity in all departments. Or find the nearby departments by this department-similarity function:

$$SIM_{dept}(D_1, D_2) = \frac{\mathrm{div}(D_1) + \mathrm{div}(D_2) - \mathrm{div}(D_1 \cup D_2)}{\mathrm{div}(D_1 \cup D_2)}.$$

Now you can compare pairs of departments, why not measure the demographic diversity of the entire university by using soft cardinality again? Now, you can obtain a university-similarity function too. Just keep going!

Consider the hypothesis that the demographic diversity of two affine collaborating departments in different institutions could lead to a Nobel Prize. Let's gather some pairs of collaboration departments that were and weren't awarded with the Nobel Prize. Now, it's possible to extract some cardinality-based features from each pair like $\frac{div(D_1)}{\mathrm{div}(D_1 \cup D_2)}$ and some others among the 272 features proposed in Section 5.2.2. Just identify some useful features and give that data to a SVM and check if the features can predict some Nobel laureates accurately. If so, you can suggest some promising collaborations.

6.3 Summary of Contributions

In this last section, we present all contributions and products related directly or indirectly to the development of this dissertation.

6.3.1 Articles in international conferences

- Our first conference article proposing the measure presented in equation 2.5 in CICLING'09, Mexico D.F., Mexico [**?**].
- The first publication about soft cardinality presented in SPIRE'10, Los Cabos, Mexico [**?**].

- Soft cardinality applied to name matching and information retrieval tasks presented in MICAI'11 [?] obtaining the best paper award.
- Contribution (co-author) to the article titled "*Towards User Profile-based Interfaces for Exploration of Large Collections of Items*" presented in the workshop Decisions@Recsys'13 in The ACM Conference on Recommender Systems 2013, Hong Kong.

6.3.2 Articles in Journals

- An article in *Applied and Computational Mathematics Journal* (A2 Colciencias ranking, IF=0.697, 2013) applying soft cardinality to name matching, information retrieval, textual entailment and paraphrase detection [?].
- Article based on the Chapter 4 of this was accepter for pubblication at *Polibits* journal (A1 Colciencias ranking).
- Article based on the Chapter 3 of this dissertation submitted to *Information Sciences* (A1 Colciencias ranking, IF=3.893, 2014).
- Article based on the Chapter 5 of this dissertation submitted to *Information Sciences.*
- Article based on the Chapter 2 of this dissertation submitted to *Computación y Sistemas* journal (A1 Colciencias ranking).
- The article titled "*BM25-CTF: Improving TF and IDF factors in BM25 by using collection term frequencies*" co-authored with Silviu Cucerzan from Microsoft Research, Redmond, WA, submitted to *Information Sciences.*

6.3.3 Participations in international competitions

- *Semantic text similarity* task at SemEval 2012, Montreal, CA [?].
- *Cross-lingual textual entailment for content synchronization* task at SemEval 2012, Montreal, CA [?].
- *Shared task, semantic textual similarity* at SemEval 2013, Atlanta, Georgia, USA [?].
- C*ross-lingual textual entailment for content synchronization* task at SemEval 2013, Atlanta, Georgia, USA [?].
- *The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge* at SemEval 2013, Atlanta, Georgia, USA [?].
- *Evaluating Phrasal Semantics* task at SemEval 2013, Atlanta, Georgia, USA [?].

- *Author Profiling* task at CLEF 2013, Valencia, Spain [?].
- *Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Entailment* at SemEval 2014, Dublin, Ireland [?].
- *Cross-Level Semantic Similarity* tasks at SemEval 2014, Dublin, Ireland [?].
- *Multilingual Semantic Textual Similarity* at SemEval 2014, Dublin, Ireland [? ?].
- *L2 (second language) Writing Assistant* task at SemEval 2014, Dublin, Ireland [?]
- *Analysis of Clinical Text* task at SemEval 2014, Dublin, Ireland [?].

6.3.4 Contributions to NLP for education

Given the success of our participation in *The Joint Student Response Analysis* task at SemEval 2013 we were motivated to collaborate in the following works:

- Conference article titled "*Automatically Assessing Children's Writing Skills Based on Age-Supervised Datasets*", presented at CICLING 2014, Katmandu, Nepal [?].
- Conference article titled "*Automatic prediction of item difficulty for short-answer questions*" submitted to 10CCC'15 (The Tenth Colombian Congress of Computation), Bogotá, Colombia.
- Conference poster article titled "*ZETEMA: A Web Service for Automatic Short-Answer Questions Grading*" submitted to 10CCC'15 (The Tenth Colombian Congress of Computation), Bogotá, Colombia.

6.3.5 Statistics of citations to our work

Until June 19th, 2015.

- H-index in Google Scholar: 5
- Total number of articles: 20
- Total number of citations in Google Scholar: 101
- Citations to our work: 47
- Citations using or extending our work: 8
- Missing: 3
- Erroneous: 14

- Self-citations: 31
- Real H-index: 4 (removing 'self' and 'erroneous')
- Total number of citations: 90 (adding 'missing' and removing 'erroneous')
- Real number of citations: 59 ('total' removing 'self')

Bibliography

[] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA, 2009. ACL. ISBN 978-1-932432-41-1.

[] Eneko Agirre, Daniel Cer, Mona Diab, and Gonzalez-Agirre Aitor. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 385–393, Montreal,Canada, 2012. ACL.

[] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 Shared Task: Semantic Textual Similarity, including a Pilot on Typed-Similarity. pages 32–43, Atlanta, Georgia, USA, 2013. ACL.

[] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2014 Task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, 2014. ACL.

[] Eneko Agirre, Carmen Banea, and others. SemEval-2015 task 2: Semantic textual similarity, English, S-panish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), June*, 2015.

[] Aletras, Nikolaos and Stevenson, Mark. A Hybrid Distributional and Knowledge-based Model of Lexical Semantics. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*. Association for Computational Linguistics, 2015.

[] Alberto Apostolico and Concettina Guerra. The longest common subsequence problem revisited. *Algorithmica*, 2(1-4):315–336, 1987.

[] B. De Baets, H. De Meyer, and H. Naessens. A class of rational cardinality-based similarity measures. *Journal of Computational and Applied Mathematics*, 132(1):51–69, July 2001. ISSN 0377-0427.

[] Baker, Simon, Reichart, Roi, and Korhonen, Anna. An Unsupervised Model for Instance Level Subcategorization Acquisition. In *Proceedings of EMNLP*, 2014.

[] Satanjeev Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, number 2276 in Lecture Notes in Computer Science, pages 136–145. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-43219-7, 978-3-540-45715-2.

[] Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval *SEM 2012)*, Montreal, Canada, 2012. Association for Computational Linguistics.

[] Baroni, Marco, Dinu, Georgiana, and Kruszewski, Germán. Don t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, ACL, 2014.

[] Montserrat Batet, David Sánchez, and Aida Valls. An ontology-based measure to compute semantic similarity in biomedicine. *Journal of biomedical informatics*, 44(1):118–125, 2011.

[] Bengio, Yoshua. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 2003(3):1137–1155, 2003.

[] Mikhail Bilenko and Raymond J. Mooney. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report AI-02-296, Artificial Intelligence Lab, University of Texas at Austin, 2002.

[] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435. doi: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993. ACM ID: 944937.

[] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. In *www*, volume 7, pages 757–766, 2007.

[] Bouma, Gerlof. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, pages 31–40, 2009.

[] Bruni, Elia, Uijlings, Jasper, Baroni, Marco, and Sebe, Nicu. Distributional semantics with eyes: using image analysis to improve computational representations of word meaning. In *Proceedings of the 20th ACM international conference on Multimedia (MM'12)*, pages 1219–1228, 2012.

[] John A. Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526, 2007.

[] Peter Christen. A Comparison of Personal Name Matching: Techniques and Practical Issues. In *Data Mining Workshops, International Conference on*, pages 290–294, Los Alamitos, CA, USA, 2006. IEEE Computer Society. ISBN 0-7695-2702-7.

[] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March 1990. ISSN 0891-2017.

[] Rudi Cilibrasi and Paul Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[] William W Cohen, Pradeep Ravikumar, and Stephen E Fienberg. A Comparison of String Metrics for Matching Names and Records. volume 3, pages 73–78, 2003.

[] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[] D. Croce, V. Storch, P. Annesi, and R. Basili. Distributional Compositional Semantics and Text Similarity. In *Proceedings of the IEEE Sixth International Conference on Semantic Computing (ICSC)*, pages 242–249, September 2012. doi: 10.1109/ICSC.2012.63.

[] Danilo Croce, Valerio Storch, and Roberto Basili. UNITOR-CORE TYPED: Combining Text Similarity and Semantic Filters through SV Regression. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: SemanticTextual Similarity*, pages 59–65, Atlanta, Georgia, USA, 2013. ACL.

[] Fred J. Damerau. A Technique for Computer Detection and Correction of Spelling Errors. *Commun. ACM*, 7(3):171–176, March 1964. ISSN 0001-0782. doi: 10.1145/363958.363994.

[] Maria De-Arteaga, Sergio Jimenez, George Duenas, Sergio Mancera, and Julia Baquero. Author profiling using corpus statistics, lexicons and stylistic features. *Online Working Notes of the 10th PAN Evaluation Lab on Uncovering Plagiarism, Authorship. and Social Misuse, CLEF*, 2013.

[] B. De-Baets, H. De-Meyer, and H. Naessens. A class of rational cardinality-based similarity measures. *Journal of Computational and Applied Mathematics*, 132(1):51–69, 2001. ISSN 0377-0427. doi: 10.1016/S0377-0427(00)00596-3.

[] B De-Baets, S Janssens, and H De-Meyer. On the transitivity of a parametric family of cardinality-based similarity measures. *International Journal of Approximate Reasoning*, 50(1):104–116, January 2009. ISSN 0888613X. doi: 10.1016/j.ijar.2008.03.006.

[] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D. Manning, and others. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. ISSN 1097-4571. doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[] Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, 1945.

[] Drucker, Harris, Burges, Chris J.C., Kaufman, Linda, Smola, Alex, and Vapnik, Vladimir. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.

[] Dueñas, George, Jimenez, Sergio, and Julia, Baquero. Automatic prediction of item difficulty for short-answer questions. In *to appear*, 2015.

[] Myroslava O. Dzikovska, Rodney D. Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (*SEM 2013)*, pages 263–274, Atlanta, Georgia, USA, 2013. ACL.

[] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate Record Detection: A Survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, 2007.

[] Stefan Evert. Corpora and collocations. *Corpus Linguistics. An International Handbook*, 2:223–233, 2008.

[] Fellbaum, Christiane, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[] Finkelstein, Lev, Gabrilovich, Evgeniy, Matias, Yossi, Rivlin, Ehud, Solan, Zach, Wolfman, Gadi, and Ruppin, Eytan. Placing search in context: the concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

[] Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[] Georg, Cantor. Ueber eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen. *J. Reine Angew. Math.*, 77:258–262, 1874.

[] Osamu Gotoh. An improved algorithm for matching biological sequences. *Journal of molecular biology*, 162(3):705–708, 1982.

[] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[] Halawi, Guy, Dror, Gideon, Gabrilovich, Evgeniy, and Koren, Yehuda. Large-scale learning of word relatedness with constraints. In *Proceedings of KDD*, pages 1406–1414, 2012.

[] M. O. Hill. Diversity and Evenness: A Unifying Notation and Its Consequences. *Ecology*, 54(2):427, March 1973.

[] Hill, Felix, Cho, KyungHyun, Jean, Sebastien, Devin, Coline, and Bengio, Yoshua. Not All Neural Embeddings are Born Equal, 2014.

[] Hill, Felix, Reichart, Roi, and Korhonen, Anna. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation, 2014.

[] Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998.

[] Huang, Eric H., Socher, Richard, Manning, Christopher D., and Ng, Andrew Y. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, volume 1, pages 873–882. ACL, 2012.

[] Paul Jaccard. Etude comparative de la distribution florare dans une portion des {A}lpes et des {J}ura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, pages 547–579, 1901.

[] Mario Jarmasz and Stan Szpakowicz. Roget's Thesaurus and Semantic Similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111, 2004.

[] M.A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84:414–420, June 1989.

[] S. P. Jena, S. K. Ghosh, and B. K. Tripathy. On the theory of bags and lists. *Information Sciences*, 132(1-4):241–254, 2001.

[] Jay J Jiang and David W Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. Taiwan, September 1997. In the Proceedings of ROCLING X, Taiwan, 1997.

[] Sergio Jimenez. *A Knowledge-Based information Extraction Prototype for Data-Rich Documents in the Information Technology Domain.* PhD thesis, Universidad Nacional de Colombia (National University of Colombia), 2008.

[] Sergio Jimenez and Alexander Gelbukh. SC Spectra: A Linear-Time Soft Cardinality Approximation for Text Comparison. In *Advances in Soft Computing*, volume 7095 of *Lecture Notes in Computer Science*, pages 213–224. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-25329-4.

[] Sergio Jimenez and Alexander Gelbukh. Baselines for Natural Language Processing Tasks. *Appl. Comput. Math.*, 11(2):180–199, 2012. ISSN 1683-3511.

[] Sergio Jimenez, Claudia Becerra, Alexander Gelbukh, and Fabio Gonzalez. Generalized Mongue-Elkan Method for Approximate Text String Comparison. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, number 5449 in Lecture Notes in Computer Science, pages 559–570. Springer Berlin Heidelberg, January 2009. ISBN 978-3-642-00381-3, 978-3-642-00382-0.

[] Sergio Jimenez, Fabio Gonzalez, and Alexander Gelbukh. Text Comparison Using Soft Cardinality. In Edgar Chavez and Stefano Lonardi, editors, *String Processing and Information Retrieval*, volume 6393 of *LNCS*, pages 297–302. Springer, Berlin, Heidelberg, 2010.

[] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. Soft Cardinality: A Parameterized Similarity Function for Text Comparison. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 449–453, Montreal, Canada, 2012. ACL.

[] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. Soft Cardinality+ ML: Learning Adaptive Similarity Functions for Cross-lingual Textual Entailment. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 684–688, Montreal, Canada, 2012. ACL.

[] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. SOFTCARDINALITY-CORE: Improving Text Overlap with Distributional Measures for Semantic Textual Similarity. In *Second Joint*

*Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 194–201, Atlanta, Georgia, USA, June 2013. ACL.

[] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. SOFTCARDINALITY: Learning to Identify Directional Cross-Lingual Entailment from Cardinalities and SMT. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 34–38, Atlanta, Georgia, USA, June 2013. ACL.

[] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. SOFTCARDINALITY: Hierarchical Text Overlap for Student Response Analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 280–284, Atlanta, Georgia, USA, 2013. ACL.

[] Sergio Jimenez, George Duenas, Julia Baquero, and Alexander Gelbukh. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, Dublin, Ireland, 2014. ACL.

[] Jimenez Sergio, Becerra Claudia, and Gelbukh Alexander. UNAL: Discriminating between Literal and Figurative Phrasal Usage Using Distributional Statistics and POS tags. Atlanta, Georgia, USA, June 2013.

[] Jimenez, Sergio, Gonzalez, Fabio A., and Gelbukh, Alexander. Cardinality-based lexical similarity in WordNet: Bridging the gap to neural embedding. *to appear*, 2015.

[] Jimenez, Sergio, Gonzalez, Fabio A., and Gelbukh, Alexander. Mathematical properties of Soft Cardinality: Enhancing Jaccard, Dice and cosine similarity measures with element-wise distance. *to appear*, 2015.

[] David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. SemEval-2014 Task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 17–26, Dublin, Ireland, 2014. ACL.

[] H. Keskustalo, A. Pirkola, K. Visala, and E. Leppanen. Non-adjacent digrams improve matching of cross-lingual spelling variants. In *LNCS 2857*, pages 252–265, Manaus, Brazil, 2003.

[] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[] Claudia Leacock and Martin Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. In *WordNet: an electronic lexical database*, pages 305–332. Fellbaum C., 1998.

[] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32, College Park, Maryland, 1999. ACL. doi: 10.3115/1034678.1034693.

[] Michael D. Lee, B.M. Pincombe, and Matthew Welsh. An empirical evaluation of models of text document similarity. In *In CogSci2005*, pages 1254–1259. Erlbaum, 2005.

[] Lee, Daniel and Seung, Sebastian. Algorithms for Non-negative Matrix Factorization. In *Advances in neural information processing systems NIPS*, pages 556–562, 2001.

[] Samuel P. Leeman-Munk, Eric N. Wiebe, and James C. Lester. Assessing elementary students' science competency with text analytics. In *Proceedins of the Fourth International Conference on Learning Analytics And Knowledge (LAK 14)*, pages 143–147, Indianapolis, Indiana, USA, 2014. ACM.

[] Tom Leinster and Christina A. Cobbold. Measuring diversity: the importance of species similarity. *Ecology*, 93(3):477–489, 2012.

[] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[] Li, Changliang, Xu, Bo, Wu, Gaowei, Zhuang Tao, Wang, Xiuying, and Ge, Wendong. Improving Word Embeddings via Combining with Complementary Languages. In *Proceedings of Canadian AI*, pages 313–318. Springer, LNAI 8436, 2014.

[] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8, 2004.

[] Dekang Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc., 1998. ISBN 1-55860-556-8.

[] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia*. Association for Computational Linguistics, 2002.

[] Luong, Minh-Thang, Socher, Richard, and Manning, Christopher D. Better word representations with recursive neural networks for morphology. In *CoNLL-2013*. ACL, 2013.

[] André Lynum, Partha Pakray, Björn Gambäck, and Sergio Jimenez. NTNU: Measuring Semantic Similarity with Sublexical Feature Representations and Soft Cardinality. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 448–453, Dublin, Ireland, 2014. ACL.

[] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland, 2014. ACL.

[] Bridget T. McInnes, Ted Pedersen, Ying Liu, Genevieve B. Melton, and Serguei V. Pakhomov. U-path: An undirected path-based measure of semantic similarity. In *AMIA Annual Symposium Proceedings*, volume 2014, page 882. American Medical Informatics Association, 2014.

[] Kenta Mikawa, Tomoyuki Ishida, and Masayuki Goto. A proposal of extended cosine measure for distance metric learning in text classification. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1741–1746. IEEE, 2011.

[] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[] Miller, George A. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

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

[] Alvaro E. Monge and Charles Elkan. The field matching problem: Algorithms and applications. In *Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 267–270, Portland, OR, 1996.

[] Erwan Moreau, François Yvon, and Olivier Cappé. Robust similarity measures for named entities matching. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 593–600, Manchester, United Kingdom, 2008. ACL. ISBN 978-1-905593-44-6.

[] Nelly Moreno, Sergio Jimenez, and Julia Baquero. Automatically Assessing Children's Writing Skills Based on Age-Supervised Datasets. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, number 8404 in Lecture Notes in Computer Science, pages 566–577. Springer Berlin Heidelberg, April 2014. ISBN 978-3-642-54902-1, 978-3-642-54903-8.

[] Gonzalo Navarro. Multiple Approximate String Matching by Counting. In *In Proc.of the 4th South American Workshop on String Processing (WSP'97)*, pages 125–139. Carleton University Press, 1997.

[] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970. ISSN 0022-2836. doi: 10.1016/0022-2836(70)90057-4.

[] Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2012. Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 399–407, Montreal, Canada, 2012. ACL.

[] Matteo Negri, Alessandro Marchetti, Yashar Mehdad, and Luisa Bentivogli. Semeval-2013 Task 8: Cross-lingual Textual Entailment for Content Synchronization. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 25–33, Atlanta, Georgia, USA, 2013. ACL.

[] Hoa Nguyen, Hisham Al-Mubaid, and others. New ontology-based semantic similarity measure for the biomedical domain. In *Granular Computing, 2006 IEEE International Conference on*, pages 623–628. IEEE, 2006.

[] Ochiai, Akira. Zoogeographical studies on the soleoid fishes found Japan and its neighboring regions. *Jap. Soc. Sci. Fish.*, 22(9):526–530, 1957.

[] Patwardhan, Siddharth and Pedersen, Ted. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, 2006.

[] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity: measuring the relatedness of concepts. In *Proceedings HLT-NAACL–Demonstration Papers*, Stroudsburg, PA, USA, 2004. ACL.

[] Pedregosa, Fabian, Varoquaux, Gaël, Gramfort, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier, Blondel, Mathieu, Prettenhofer, Peter, Weiss, Ron, Doubourg, Vincent, Vanderplas, Jake, Passos, Alexandre, Cournapeau, David, Brucher, Matthieu, Perrot, Matthieu, and Duchesnay, Edouard. Scikit-learn: Machine Learning in {P}ython. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12, pages 1532–1543, Doha, Qatar, 2014.

[] Ari Pirkola, Heikki Keskustalo, Erkka Lepp{\"a}nen, Antti-Pekka K{\"a}ns{\"a}l{\"a}, and Kalervo J{\"a}rvelin. Targeted s-gram matching: a novel n-gram matching technique for cross-and monolingual word form variants. *Information research*, 7(2):7–2, 2002.

[] Riccardo Poli, William B. Langdon, Nicholas F. McPhee, and John R. Koza. *A field guide to genetic programming*. Lulu. com, 2008.

[] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169. ACM, 1993.

[] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.

[] Radinsky, Kira, Agichtein, Eugene, Gabrilovich, Evgeniy, and Markovitch, Shaul. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web , WWW '11*, pages 337–346, NY, USA, 2011. ACM.

[] A. Renyi. On measures of entropy and information. In J. Neyman, editor, *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561, Berkeley, 1961.

[] Philip Resnik. Using Information Content to Evaluate Semantic Similarity. In *Proceedings of the 14th International Joint Conference on artificial Intelligence*, Montreal, Canada, 1995.

[] Phillip Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[] Carlo Ricotta and Laszlo Szeidl. Towards a unifying approach to diversity measures: Bridging the gap between the Shannon entropy and Rao's quadratic index. *Theoretical Population Biology*, 70(3):237–243, November 2006.

[] Eric S. Ristad and Peter N. Yianilos. Learning string edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[] Alejandro Riveros, Maria De-Arteaga, Fabio A. González, and Sergio Jimenez. MindLab-UNAL: Comparing Metamap and T-mapper for Medical Concept Extraction in SemEval 2014 Task 7. *SemEval 2014*, page 424, 2014.

[] Alexander M. Robertson and Peter Willett. Applications of n-grams in textual information systems. *Journal of Documentation*, 54(1):48–67, 1998.

[] S. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994)*, pages 109–126, Gaithersburg, USA, 1994.

[] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, October 2004. ISSN 0022-0418.

[] Peter Mark Roget. *Roget's Thesaurus of English Words and Phrases...* TY Crowell Company, 1911.

[] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965. ISSN 0001-0782. doi: 10.1145/365628.365657.

[] Rubinstein, Herbert and Goodenough, John B. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[] M Sahlgren. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, 2005.

[] Gerard Salton. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[] Gerard Salton, Andrew K. C. Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[] Hinrich Schütze. Dimensions of meaning. In *Supercomputing'92., Proceedings*, pages 787–796. IEEE, 1992.

[] C. E. Shannon. A Mathematical Theory of Communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001. ISSN 1559-1662.

[] Claude E. Shannon. Prediction and entropy of printed English. *Bell system technical journal*, 30(1):50–64, 1951.

[] Grigori Sidorov, Alexander Gelbukh, Helena Gomez-Adorno, and David Pinto. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computacion y Sistemas*, 18(3):491–504, 2014.

[] Emilio Silva-Schlenker, Sergio Jimenez, and Julia Baquero. UNAL-NLP: Cross-lingual phrase sense disambiguation with syntactic dependency trees. *SemEval 2014*, page 743, 2014.

[] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, March 1981. ISSN 0022-2836. doi: 10.1016/0022-2836(81)90087-5.

[] Kent A. Spackman, Keith E. Campbell, and Roger A. Côté. SNOMED RT: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium*, page 640. American Medical Informatics Association, 1997.

[] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. ISSN 0022-0418.

[] Szumlanski, Sean, Gomez, Fernando, and Sims, Valerie K. A New Set of Norms for Semantic Relatedness Measures. In *ACL'13*, pages 890–895, 2013.

[] Taieb, Mohamed Ali Hadj, Aouicha, Mohamed Ben, and Hamadou, Abdelmajid Ben. Computing semantic relatedness using Wikipedia features. *Knowledge-Based Systems*, 50:260–278, September 2013.

[] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. ISSN 0033-295X. doi: 10.1037/0033-295X.84.4.327.

[] Esko Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191–211, January 1992. ISSN 0304-3975. doi: 10.1016/0304-3975(92)90143-4.

[] Julian R. Ullmann. A Binary N-Gram Technique for Automatic Correction of Substitution, Deletion, Insertion and Reversal Errors in Words. *The Computer Journal*, 20(2):141–147, January 1977. ISSN 0010-4620, 1460-2067. doi: 10.1093/comjnl/20.2.141.

[] Jorge A. Vanegas, Juan C. Caicedo, Jorge E. Camargo, and Raul Ramos-Pollán. Bioingenium at ImageCLEF 2012: Textual and Visual Indexing for Medical Images. In *CLEF (Online Working Notes/Labs/Workshop)*, Rome, Italy, 2012.

[] X. Wan. A novel document similarity measure based on earth mover's distance. *Information Sciences*, 177(18):3718–3730, 2007.

[] M. S Waterman, T. F Smith, and W. A Beyer. Some biological sequence metrics. *Advances in Mathematics*, 20(3):367–387, 1976. ISSN 0001-8708. doi: 10.1016/0001-8708(76)90202-4.

[] William E. Winkler. The State of Record Linkage and Current Research Problems. *Statistical Research Division, US Census Bureau*, 1999.

[] Zhibiao Wu and Martha Palmer. Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751.

[] R. Yager. Cardinality of fuzzy sets via bags. *Mathematical Modelling*, 9 (6):441–446, 1987.

[] Yang, Dongqiang and Powers, David M. W. Verb Similarity on the Taxonomy of WordNet. In *Proceedings of the Third International WordNet Conference (GWC-06)*, pages 121–128, Jeju Island, Korea, 2006.

[] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[] Zesch, Torsten, Muller, Christof, and Gurevych, Iryna. Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 861–866, 2008.