Embedding Meaning Algebra into Distributional Semantics

Author

Carlos Alonso Viñas

BSc in Physics & MSc in Theoretical Physics

Advisors

Enrique Amigó Cabrera

Víctor Fresno Fernández

Master's degree in Language Technologies

Computer Systems and Language Department

Contents

Acknowledgements

I would like to thank a number of people for the support I have received during the course of this work.

First of all, Judy, for her patience and love throughout all these weekends of work instead of going out for sushi.

To my parents and my brother for always being a refuge and a foundation for me to grow and develop myself.

To Enrique and Víctor, my advisors, for everything I have learned from them and the support they have given me, and for always keeping a close relationship with me.

And finally, to all the people with who I have been able to reflect on the work or who have helped me not to go crazy during the whole process.

Abstract

The field of distributional semantics has seen significant progress in recent years due to advancements in natural language processing techniques, particularly through the development of Neural Language Models like GPT and BERT. However, there are still challenges to overcome in terms of semantic representation, particularly in the lack of coherence and consistency in existing representation systems.

This work introduces a framework defining the relationship between a probabilistic space, a set of meanings, and a vector space of static embedding representations; and establishes formal properties based on definitions that would be desirable for any distributional representation system to comply with in order to establish a common ground between distributional semantics and other approaches. This work also introduces an evaluation benchmark, defined on the basis of the formal properties introduced, which will allow to measure the quality of a representation system.

1 Introduction

Distributional semantics has undergone a major advance thanks to the existence of large amounts of data, the increase in computational processing capabilities and advances in natural language processing (NLP) techniques, especially thanks to the development of Neural Language Models such as the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) or the Generative Pretrained Transformer (GPT) (Radford et al., 2018, 2019) and Large Language Models.

However, despite this huge progress, there are still gaps to address in terms of semantic representation. Semantic representation has important implications in areas such as machine translation, information retrieval, the correction of non-semantic spaces by contrastive methods and natural language understanding. Additionally, researchers in both NLP and cognitive science are working to establish parallelisms in the way humans learn a language and how large language models acquire knowledge from their training data, increasing the interest on how these models encode and represent the knowledge with the aim of being able to shed light in how humans acquire and represent the same knowledge.

One of the most notable problems is the lack of coherence and consistency in the representation of words and their semantic relations in existing representational systems (Maruyama, 2019). The main reason is that distributional semantics focuses on the study of how words acquire meaning through their use in specific contexts (the principle of contextuality), as opposed to other approaches based on formal definitions and grammatical categories (following the principle of compositionality).

In this context, and in order to establish some common elements between both approaches, we propose a theoretical framework based on a semantic space (a space of meanings) ruled by a set algebra, as well as its connection with the vector space of embeddings and a space of word sequences. This allows the establishment of a set of formal rules that allow a coherent and consistent representation of words and their semantic relations.

The rest of this work is organized into five sections: Section 2 focuses on the motivation behind the work, highlighting the specific challenges. In Section 3 a theoretical framework is introduced, laying the foundation for the subsequent analysis and experimentation. Section 4 outlines the methodology employed to create a benchmark, which serves as a reference point for evaluating the benchmark. Section 5 presents the experimental findings, discussing the various proposals and contrasting their results of the benchmark testing. Finally, Section 6 offers conclusions, summarizing the key contributions, drawing meaningful insights from the study, and suggesting future lines of research.

2 Related work and motivation

There are different approaches and lines of research within distributional semantics. On one hand, most of the work on evaluation of models based on distributional semantics focuses on the effectiveness of the systems, either in user-oriented tasks (classification, dialogue, etc.) (Bailly and Leblond, 2023; Chen et al., 2023; Dar et al., 2022; Shi et al., 2022; Su et al., 2021; Baroni, 2020; Belinkov and Glass, 2019) or in testing linguistic capabilities (probes) (Alain and Bengio, 2016; Linzen et al., 2016; Giulianelli et al., 2018). This line is also working in adding a compositional (semantic) layer to distributional representation systems, with the objective of achieving Compositional Distributional Semantic models (Clark and Pulman, 2007; Mitchell and Lapata, 2008; Coecke et al., 2010; Arora et al., 2017; Bastings et al., 2018; Valvoda et al., 2022).

On the other hand, there are approaches that address the problem of representation of word sequences regardless of the task to which it is applied: how to represent complex texts and the analysis of the relations and operators of these representations. This second line, focused on representation, can itself be divided into three different types of studies, that focus on one aspect of the embedding process and subsequent manipulation, trying to study and modify it in order to add that semantic layer: studies on embedding functions, studies on composition functions and studies on similarity functions and text distribution in embedding spaces.

2.1 Studies on embedding functions

In the seminal Mikolov's work, words are mapped to vectors (Mikolov et al., 2013a,b) by trying to predict words given another words. These models are trained by sliding a window along a text corpus and using the central word of the window to predict the words of its vicinity. These approach created static embeddings. Some extensions proposed work on representing longer linguistic units (Kiros et al., 2015; Le and Mikolov, 2014).

Newer models are more contextual. Models sensitive to word order, such as LSTMs (Hochreiter and Schmidhuber, 1997) and variations such as CoVe (McCann et al., 2017) or ELMo (Peters et al., 2018), introduced a memory cell to preserve a state over long periods of time; while graph based models, such as Transformers (Vaswani et al., 2017) and its derivations such as BERT (Devlin et al., 2019), BART (Lewis et al., 2019) or GPT (Brown et al., 2020), use a fully connected graph to model the relations of every words in the input text. Transformers can solve several downstream tasks from a limited training set via fine-tuning, and have great predicting power over word strings as Language Models (Radford et al., 2019). However, contextual models do not preserve the isometry between the representation of words and their meanings. This is known as the representation degradation problem (Ethayarajh, 2019; Gao et al., 2019a; Hupkes et al., 2020).

2.2 Studies on composition functions

In the field of compositional distributional text representation, some other works focus on composition functions that allow generating complex representations from simple ones (e.g., words). A large body of literature has shown that the sum or global average of word embeddings is very effective, often outperforming more sophisticated methods (Mitchell and Lapata, 2010; Boleda, 2020; Lenci, 2018; Blacoe and Lapata, 2012; Perone et al., 2018; Baroni and Lenci, 2010; Rimell et al., 2016; Czarnowska et al., 2019; Wieting and Gimpel, 2018; Ethayarajh, 2018). However, additive approaches do not consider word order and their effectiveness degrades with sequence length (Polajnar et al., 2014).

Clarke et al. (2011) proposed several algebraic approaches to compose meaning. The approaches include quotient algebras, finite algebras and algebras derived from semigroups. Algebraic approaches also include the work of Clark and Pulman (2007), that introduced a tensor product as composition function, but space complexity grows exponentially; and Smolensky et al. (2016), that worked on mapping inference in predicate logic-based on tensor product representations, but a previous mapping between sentences and logical propositions is required. Coecke et al. (2010) proposed a composition function based on algebra of pre-groups, proposing dot product as an approach to similarity.

Other authors have linked their compositional functions with Information Theory.

Arora et al. (2017) applied IDF (word specificity, which has a direct correspondence with the Information Content from Shannon's Information Theory) and Singular Value Decompositon to achieve similar results as sequential models such as LSTMs. Amigó et al. (2022) introduced a new composition and similarity function based on Shannon's Information Theory that generalizes traditional approaches and fulfills formal properties previously established. This function considers

2.3 Studies on similarity functions and text distribution

Other studies focus on measuring and interpreting the semantic proximity of text representations in the embedding space. On one hand, there is a significant literature on semantic similarity among representations. Some traditional similarity functions between representations are the cosine distance, euclidean distance, dot product or, more recently, the Information Contrast Model (Amigó et al., 2020).

On the other hand, there is also literature that focuses on studying how representations are distributed in space. Ethayarajh (2019) researched about how contextual are contextual word embeddings, finding anisotropy in all the studied models and its impact in similarity measures, and Baggetto and Fresno (2022) analyzed if the space anisotropy of contextual models spaces is the reason why they underperform in semantic tasks with respect to static embeddings, finding no clear correlation between isotropy and semantic isometry.

2.4 Situating our work

In this work we start from a holistic perspective. That is, one should not study the **embedding function**, **composition**, or **similarity function** independently, as all of these components together are interconnected and form a system. Hence, we propose the development of an evaluation framework to compare models based on their internal coherence among all these components.

3 Theoretical framework: Representation Systems

This section introduces the foundational principles of Distributional Semantic Representation Systems. We first analyze the different distributional semantic paradigms, and then we introduce different spaces and functions to define Distributional Semantic Representation Systems. After this, we define the properties and constraints that these functions must follow.

3.1 Distributional Semantic Paradigms

We can distinguish four distributional semantic paradigms in natural language processing. Figure 1 represents the first of them, which we will call the **decision learning paradigm**: the machine receives an expression or text in natural language and makes decisions based on it.

In this paradigm there are three spaces: that of *word sequences*, which is the one we humans use to communicate; a *vector space* in which the machine represents the word sequences it receives; and a *meaning space* that is reduced to a small set of categories such as true or false, spam or non-spam, positive or negative opinion, topics or some named entities.

The fact of translating the sequences into a continuous vector space allows the machine to make decisions in the face of new stimuli by proximity in space. This mode of vector representation has been in place since the early years of machine learning. Text classifiers, web search engines, extractive summaries, etc. are found at this level. The main limitation of this paradigm is that the logic decision space is very limited and conditioned by the amount of categories that can be learned by a set of training samples, and by how these samples are distributed within the representation space.

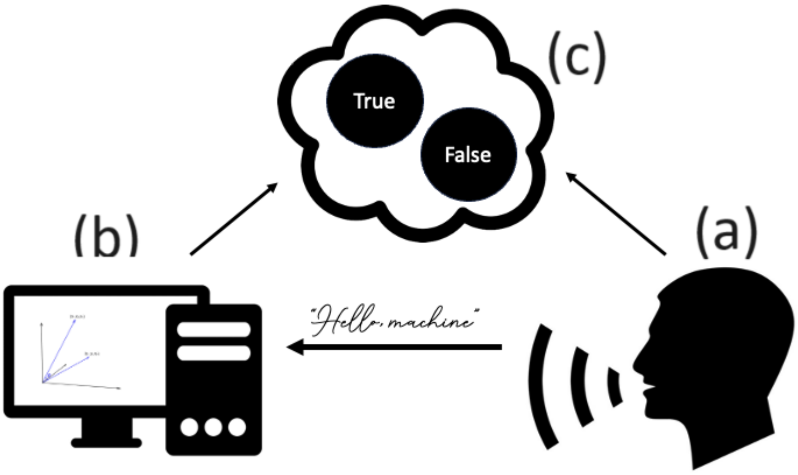

Figure 1: Decision Learning paradigm. The machine receives an expression or text in natural language (*word sequences*) (a), represents it in a *vector space* (b) and makes decisions contained in a *meaning space* (c).

In the second paradigm, called **Generative Language Modeling paradigm** (see Figure 2), the machine does not makes decisions but is able to generate language, responsing words to which a probability is assigned given an input sequence. This allows for automatic language generation and gives the impression of some creativity on the part of the machine. This type of systems started with automatic translators and have evolved into today's conversational systems, including the famous ChatGPT (OpenAI, 2022) and its main competitors, such as Bard (Google, 2023) or LLAMA (Meta, 2023).

The strengths of this paradigm are that word prediction can be trained over huge text collections without needing human annotation.

The recent advance of these systems is due to deep neural language models, which interconnect the *probabilistic space of word sequences* (language models) with the *vector space*. The word sequences are represented, after a pre-training stage, by activating input and output neurons, and the vector space is represented by the internal state of the network. The key point is that this paradigm mixes the word prediction power of statistical language models with the generalization power of vector representations. Therefore, the system is able to give responses to unseen utterances (i.e., word sequences).

The limitation of this paradigm is that the system is not able to make precise decisions but generating words. The problem grows when we deal with specific domains (e.g. a small business), for which there is not enough text to generate accurate and precise answers by language models alone. Notice that a conversational system, however powerful it may be, does not have a semantic model of the world, and knows how to speak but does not know what is saying.

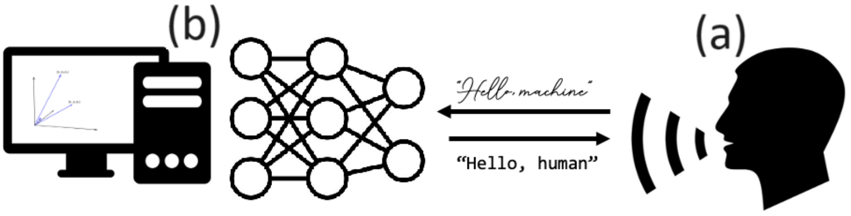

Figure 2: Generative Language Modeling paradigm. Recent neural language models interconnect the *probabilistic space of word sequences* (a) with the *vector space* (b). The word sequences are represented by activating input and output neurons. The vector space is represented by the internal state of the network.

In order to mitigate this limitation, the third paradigm, called **Discriminative Language Modeling Paradigm** (see Figure 3), consists in adding neuron layers for training again the pre-trained language model with a few annotated samples. This is commonly referred to as fine-tuning or few shot learning (Wang et al., 2020; Dodge et al., 2020; Liu et al., 2022).

The pre-trained step (language modeling) improves the generalization power provided by the original decision learning paradigm and the required number of training samples is considerably reduced. The limitation remains that the field of accurate decisions (true/false) is still small.

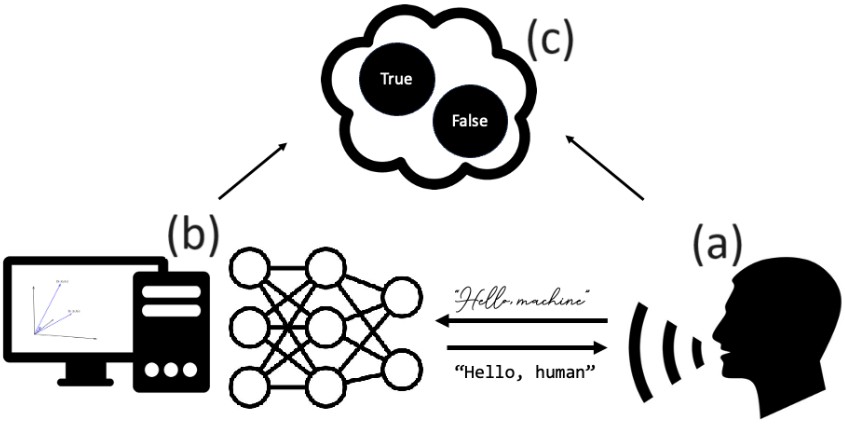

Figure 3: Discriminative Language Modeling Paradigm. New neuron layers are added for doing a fine tuning step after/over the pre-trained language model. The pre-trained step improves the generalization power provided by the original decision learning paradigm and the required number of training samples is considerably reduced.

An ideal future paradigm, called **Neuronal symbolic paradigm**, should include a representation space of meanings or concepts at a logical level rich enough to make inferences and not just generate probable answers in text format (see Figure 4). This paradigm has been refereed as *Semantic Parsing*, defined as *converting natural language utterances to logical forms that can be easily executed on a knowledge base* (Kamath and Das, 2018). Chen et al. (2020) talks about *Neural Symbolic Paradigm in readers*, and Venhuizen et al. (2019) called it *Formal Distributional Semantics.*

The fundamental barrier is the machine's lack of knowledge of the world and experience. Notice that the machine experience (beyond assimilating millions of word sequences) does not go beyond the set of hand-categorised training samples. It is necessary to explicitly define a logical model of the world, making it applicable only to very specific contexts. For instance, training a language model to translate word sequences into data base queries or a programming language. Even so, there are limitations: although there are a huge set of translation samples between natural languages, each domain would require a wide set of natural/formal translation samples.

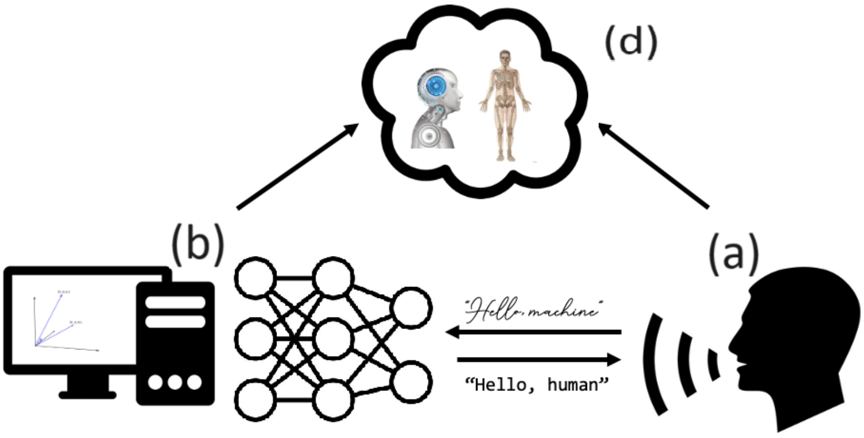

Figure 4: Neuronal symbolic paradigm. It includes a representation space of meanings or concepts (d) at a logical level rich enough to make inferences and not just generate probable answers in text format.

The paradigm proposed in this work, called **Distributional semantic algebra**, is less ambitious. The idea is that, in the absence of an extensive annotated corpus for each domain, it is very difficult to model a symbolic distributional semantics. However, **without the need for training on annotated data, and without explicitly modelling a propositional logic space, we can define basic operators on the distributional semantic representation that are consistent with the meaning space** (see Figure 5).

Specifically, in this work we study the formal and empirical properties that the operators of **text embedding functions**, **specification** and **generalization** of vector representation pairs, **similarity** between vector representations, and **meaning information cuantification** should have. In order to do so, **we consider the relation between a space of word sequences, a space of meanings and a space of embeddings**. Although the space of meanings is purely theoretical (and not computable), its properties allows us to derive the properties of the embedding, composition and similarity functions in the embedding space.

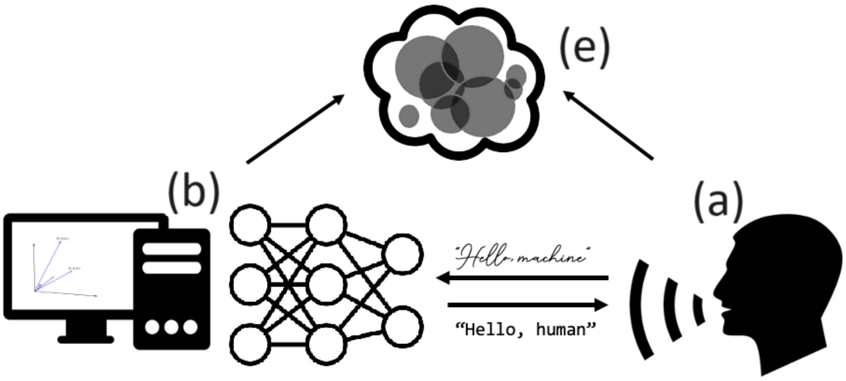

Figure 5: Distributional semantic algebra. Considering the relation between a space of word sequences (a), a space of meanings (e) and a space of embeddings (b) it is possible to obtain formal properties without the necessity of an extensive annotated corpus for each domain or explicitly modelling a propositional logic space.

3.2 Word Sequences, Embedding and Meaning Spaces

As described previously, in our theoretical model we consider three interacting spaces. First, the language model, i.e. the **space S of all possible word sequences**. In a language model, there is an infinite sample space of simple events consisting in all the possible infinite word sequences. This is a probability space in which a finite word sequence is a compound probabilistic event containing all the possible word sequences extensions. That is, the total frequency for the word 'Hello' will be higher than for the sequence 'Hello world', and hence, $P(\text{“Hello”}) > P(\text{“Hello world”}) > \ldots.$

In order to state the desirable properties of the embedding space and its associated algebraic functions, we consider a **theoretical sample space Ψ of meaning**. In this probabilistic space, each meaning is a possible outcome. Each word sequence leads to a compound event in the probabilistic meaning space. More formally, we assume the existence of a non computable function $\pi_S : S \longrightarrow \mathcal{P}(\Psi)$, from the space of word sequences S to the power set of meanings $\mathcal{P}(\Psi)$. Therefore, the more a word sequence s contains information, the more its meaning is specific and the cardinality of π_S is small as well as its probability $P(\pi_S(s))$.

It's necessary to take into account that not all possible meanings (of a word sequence) have the same degree of association with the word sequence. This aspect can be modeled by means of fuzzy events. Events are fuzzy sets of elements of the sample space (meanings), i.e., a certain membership value for each meaning bounded between 0 and 1. The probability of fuzzy events can be estimated with the Zadeh's integral (Zadeh, 1968). In the end, each word sequence is translated by the function π_S into a vector of membership values with as many dimensions as there are meanings in Ψ.

The key assumptions from which we derive the desirable properties of the embedding, aggregation, and similarity functions in the embedding space are as follows:

- **Any change on a word sequence implies a meaning change**. In other words, however small it may be, a change in the words used in an expression implies some difference in terms of their possible meanings. More formally, being $s, s' \in S$, $(s \neq s') \Longrightarrow (\pi_S(s) \neq \pi_S(s'))$.

- **The empty word sequence does not provide any information and is equally associated to all possible meanings**. Formally, $\pi_S(" \quad ") = \Psi$, i.e. all meanings have a maximum membership.

- **An infinitely precise meaning requires an infinite amount or words to be expressed**, i.e. $\lim_{P(\pi_S(s))\rightarrow 0} |s| = \infty$.
- **Every word sequence in the language model make sense at a linguistic level**. Therefore, all word sequences in S have at least a meaning with maximum membership in $\pi_S(s)$.

Finally, we consider the **embedding space** V, i.e., where word sequences are projected as n-dimensional vectors. Just like in word sequences, we assume the existence of a non computable function $\pi_V : V \longrightarrow \mathcal{P}(\Psi)$, from the space of embeddings V to the fuzzy power set of meanings $\mathcal{P}(\Psi)$.

These three spaces and their relationships are synthetized in Figure 6, together with an embedding function $\pi : S \rightarrow V$ that will be defined in the next sections. We will see that the theoretical space of meanings and the two non-computable functions that relate it to the space of word sequences and embeddings will allow us to establish the desirable properties of a representation system, particularly the properties of projection, aggregation ($\oplus$ and $\odot$), and similarity functions among embeddings.

Notice that, for our purposes, we do not need to specify some aspects such as which distribution follows the probability space Ψ or the way in which the function π_S transforms sequences of words into fuzzy sets of meanings. It is enough for us to assume that Ψ is a probability space of meanings and that the π_S output represents the specificity of word sequences in terms of possible meanings.

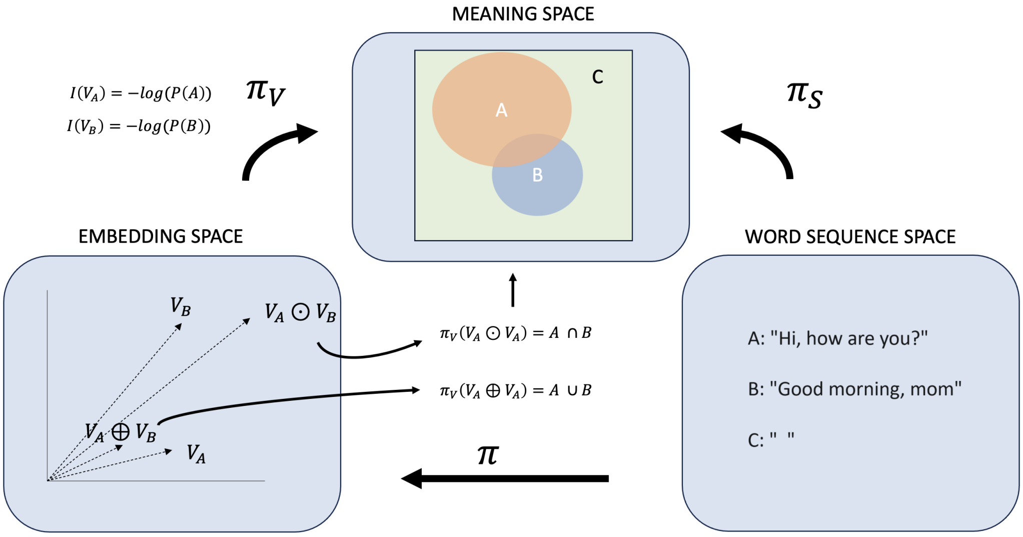

Figure 6: We consider three interacting spaces: the **space** S **of all possible word sequences**, consisting in all the possible infinite word sequences; a **theoretical sample space** Ψ **of meaning**, where each word sequence leads to a compound event in the probabilistic meaning space (not all possible meanings have the same degree of association with the word sequence, and hence must be modeled by means of fuzzy sets of elements of the sample space); and the **embedding space** V, where word sequences are projected as n-dimensional vectors. We assume the existence of a non computable function $\pi_S : S \longrightarrow \mathcal{P}(\Psi)$, from the space of word sequences S to the power set of meanings $\mathcal{P}(\Psi)$, and a non computable function $\pi_V : V \longrightarrow \mathcal{P}(\Psi)$, from the space of embeddings V to the fuzzy power set of meanings $\mathcal{P}(\Psi)$. Finally, we also work with an embedding function $\pi : S \rightarrow V$.

3.3 Definition and properties of a representation system

Generally, in the literature, when discussing a distributional representation systems, there is a talk of an embedding function that projects words or word sequences into an n-dimensional vector space. Subsequently, their properties are studied, such as how the representations are distributed in space, or what happens when they are combined using different operators like addition or dot product. However, how embeddings are distributed depends on how the similarity function is defined. Furthermore, different methods for combining embeddings can be consistent with different embedding functions. In this work, we adopt an holistic perspective. We propose that a representation system includes all these interrelated components.

Definition 1 *[Distributional Semantic Representation System] A distributional semantic representation system is a tuple* $(\pi, \mathtt{I}, \delta, \odot, \oplus)$ *of an embedding* π*, information measurement* $\mathtt{I}$*, similarity* δ*, specificity* $\odot$ *and generalization* $\oplus$ *functions:*

$$\pi : S \to V, \qquad \mathtt{I} : V \to \mathbb{R}$$

$$\delta : V \times V \to \mathbb{R}, \qquad \odot, \oplus : V \times V \to V$$

Let's now define each of the components of the representation system. The embedding function π projects word sequences into a continuous vector space (see Figure 6). In Figure 6 the word sequences *"Hi, how are you?"* and *"Good morning, mom."* are translated into two vectors V_A and V_B by the embedding function π.

Additionally, there should exist a continuous function π_V from the embedding space V to the corresponding fuzzy set in Ψ. This implies that a subtle semantic change in a word sequence should translate to a subtle change in the embedding space. In turn, the theoretical function π_V translates these vectors into meaning sets A and B, matching the output of the π_S function applied to the original word sequences.

Notice that *"Good morning, mom."* is more specific than *"Hi, how are you?"*. Since the second one is a generic greeting while the first one is a good morning greeting to a mother, the fuzzy meaning subset of *"Good morning, mom."* should cover a smaller area. Furthermore, in accordance with the assumptions described in the previous section, the empty word sequence should correspond to the complete set of meanings, that is, maximum uncertainty.

Definition 2 *[EMBEDDING FUNCTION] An embedding function $\pi : S \to V$ associates the space of word sequences S to the embedding space V in such a way that there exists a continuous function $\pi_V : V \longrightarrow \mathcal{P}(\Phi)$ from the embedding to the fuzzy power set of meanings such that:* $\forall s \in S : \quad \pi_V\Big(\pi(s)\Big) = \pi_S\Big(s\Big)$

A fundamental aspect of textual information representation is the ability to measure information. Traditionally, the amount of textual information has been measured by applying Shannon's Information Theory to the space of word sequences. However, the specificity of a word sequence doesn't always align with the specificity of its meaning. In our framework, we apply the notion of Information Quantity $(-log(P(x)))$ directly to the space of meanings. An Information Quantity measurement in the embedding space must correspond to the specificity of the fuzzy subset of meanings associated with the represented word sequence. In Figure 6 (top left equations) the information measurement function for word sequences A and B returns the minus logarithm of the corresponding probabilities in the theoretical meaning space.

Definition 3 *[Information Measurement Function] An information measurement function $\mathtt{I} : V \to \mathbb{R}$ from the embedding space to the real numbers estimates the specificity of the corresponding meaning subset. For all embedding $v \in V: \quad \mathtt{I}(v) \propto -log\Big(P(\pi_V(v)\Big)$*

Since the space Ψ is purely theoretical, as are the projection functions π_S and π_V, we cannot compute the information measurement function. However, based on this definition, we will be able to derive necessary properties that an information measure should satisfy.

Since we model the semantics of texts in terms of subsets in the space of meanings, we can now establish composition operators between texts. Specifically, the intersection of meaning sets corresponds to an *specification* operator. In particular, in Figure 6, the specification of A and B would correspond to a good morning greeting to a mother, along with asking how she is. This implies a more restricted (specific) set of possible meanings ($A \bigcap B$). Notice that the intersection in fuzzy sets is defined as the minimum membership values.

Definition 4 *[Specification] A specification function $\odot : V \times V \to V$ is a function that returns a representation which projection on the meaning space corresponds with the meaning set intersection. Formally, $\forall\ v_1, v_2 \in V: \ \pi_V(v_1 \odot v_2) = \pi_V(v_1) \cap \pi_V(v_2)$*

On the other hand, the union of meaning sets corresponds to a generalization operator. In the example from Figure 6, the union of meaning sets A and B would correspond to any greeting in which either the recipient's well-being is inquired about or a morning greeting to a mother is involved. Notice that the union in fuzzy sets is defined as the maximum membership values.

Definition 5 *[GENERALIZATION] A generalization function $\oplus : V \times V \to V$ is a function that returns a representation which projection on the meaning space corresponds with the meaning set union. Formally,* $\forall\ v_1, v_2 \in V :\ \pi_V(v_1 \oplus v_2) = \pi_V(v_1) \cup \pi_V(v_2)$

The last element of the semantic-distributional representation system is the similarity measure. To define this, we take as a reference the properties suggested by Tversky in his cognitive studies (Tversky, 1977). These studies have served as a reference for defining similarity measures in text representation (Amigó et al., 2020). Tversky's studies are particularly suitable for our context, as the author defined objects in terms of sets of features. This aligns with our representation of textual semantics in terms of sets of meanings.

In addition, Tversky's studies dismiss traditional metric properties in euclidean spaces (maximality, triangular inequality, symmetry), and states that self similarity grows with the object specificity, and that similarity is assymetric, i.e., the similarity between an specific object and its generalization is larger than vice-versa. Tversky explains this assymetricity in the choice of subject and referent based on the relative salience of the objects, as we tend to select the more salient as referent (the 'prototype') and the less salient as the subject (the 'variant'). This is why we tend to say 'the portrait resembles the person' rather than 'the person resembles the portrait'.

Definition 6 *[SIMILARITY FUNCTION] A meaning similarity function* $\delta : V \times V \to \mathbb{R}$ *is a function that satisfies:*

- *Self similarity specificity:* $P(\pi_V(v_1)) < P(\pi_V(v_2)) \Longrightarrow \delta(v_1, v_1) > \delta(v_2, v_2)$

- *Monotonicity regarding intersection and union:*

$$\left.\begin{array}{l} \pi_V(v_1) \cup \pi_V(v_2) = \pi_V(v_1) \cup \pi_V(v_3) \\ \pi_V(v_1) \cap \pi_V(v_2) > \pi_V(v_1) \cap \pi_V(v_3) \end{array}\right\} \Longrightarrow \delta(v_1, v_2) > \delta(v_1, v_3) \tag{3.1}$$

$$\left.\begin{array}{l} \pi_V(v_1) \cap \pi_V(v_2) = \pi_V(v_1) \cap \pi_V(v_3) \\ \pi_V(v_1) \cup \pi_V(v_2) < \pi_V(v_1) \cup \pi_V(v_3) \end{array}\right\} \Longrightarrow \delta(v_1, v_2) > \delta(v_1, v_3) \tag{3.2}$$

- *Assymetricity:* $\pi_V(v_1) \subset \pi_V(v_2) \Longrightarrow \delta(v_1, v_2) > \delta(v_2, v_1)$

3.4 Formal constraints

We have defined the embedding π, aggregation $\odot, \oplus$, information measurement `I`, and similarity δ functions in terms of their projection in the theoretical space of meanings Ψ. The projection functions π_V, and π_S are necessarily unknown. Since this space is therefore not computable, implementing the functions directly from their definitions is not possible. However, these definitions allow us to establish constraints on how these functions interact with each other. Based on these constraints, we will set up a set of tests for comparing different real implementations of these functions.

3.4.1 Embedding and Information Measurement functions

Regarding the embedding function π and the information measurement function $\mathtt{I}$, the first constraint refers to the sensitivity of the embedding function to small changes in word sequences. Formally,

Constraint 1 *[INJECTIVITY]*

Any difference in an expression implies a difference in meaning. Being x and y two word sequences in S: $x \neq y \longrightarrow \pi(x) \neq \pi(y)$

According to the definition of π_S (i.e. projection of word sequences into the meaning space), and the definition of the information measurement function $\mathtt{I}$, the information measurement function should return zero for the representation of the empty sequence. That is,

Constraint 2 *[ZERO INFORMATIVENESS]*

The the empty sequence representation has zero information.

Formally, $\mathtt{I}(\pi(\text{" "})) = 0$

On the contrary, we can always over-specify (with more words) a set of meanings to a subset (of lower probability) included in this set. Therefore, only an infinite amount of words can contain an infinite amount of information. More formally,

Constraint 3 *[Infinite informativeness]*

Infinite informativeness implies an infinite length sequence.

Formally, $\mathtt{I}(\pi(s)) = \infty \longrightarrow |s| = \infty$.

3.4.2 Specification and generalization functions

According to the definition of I, $\oplus$ and $\odot$, the properties of union and intersection sets, the properties of the union and intersection of fuzzy sets in Ψ, the definition of π_S and its convergence with π_V across π, we can infer that generalizing will reduce the amount of information (the more 'vague' are the meanings associated to a word sequence, the less amount of information it will contain), and that the specification will increase the amount of information. Formally, this implies that

Constraint 4 *[Generalization and Specification Information Monotonicity] The information content grows with the specification and decreases with the generalization. Formally, being* $v_1 \neq v_2 \in V$*:*

$$\mathtt{I}(v_1), \mathtt{I}(v_2) < \mathtt{I}(v_1 \odot v_2)$$

$$\mathtt{I}(v_1 \oplus v_2) < \mathtt{I}(v_1), \mathtt{I}(v_2)$$

In addition, since generalization and specification are defined in terms of union and intersection of fuzzy meaning sets in Φ, they inherit properties such as Associativity, Commutativity, Distributiveness and Idempotency:

Constraint 5 *[Commutativity]*

$$v_1 \odot v_2 = v_2 \odot v_1, \qquad v_1 \oplus v_2 = v_2 \oplus v_1$$

Constraint 6 *[Associativity]*

$$(v_1 \odot v_2) \odot v_3 = v_1 \odot (v_2 \odot v_3), \qquad (v_1 \oplus v_2) \oplus v_3 = v_1 \oplus (v_2 \oplus v_3)$$

Constraint 7 *[Distributiveness]*

$$(v_1 \oplus v_2) \odot v_3 = (v_1 \odot v_3) \oplus (v_2 \odot v_3), \quad (v_1 \odot v_2) \oplus v_3 = (v_1 \oplus v_3) \odot (v_2 \oplus v_3)$$

Constraint 8 *[Idempotency]*

$$(v_1 \oplus v_2) \oplus v_2 = v_1 \oplus v_2, \qquad (v_1 \odot v_2) \odot v_2 = v_1 \odot v_2$$

Additionally, there must be a neutral element for both operators, just like the union and intersection operators of sets. There exists a representation $v_{gen} \in V$ that is so general that its specification with any other meaning v produces the same meaning v. Similarly, there exists a representation v_{spe} that is so specific that its generalization with any other meaning v produces the same representation v. In particular the identity element of the specifiction function $\odot$ is the empty sequence representation $\pi(\text{""})$. More generally, any generalization of v (i.e. $v \oplus v'$) is a neutral element for v. On the contrary, there is no neutral element for the generalization function $\oplus$, but depends of the vector. The generalization neutral element of v is any specification of v (i.e. $v \odot v'$). Formally,

Constraint 9 *[Neutral elements]*

$$\forall v' \in V : v \oplus (v \odot v') = v \qquad \forall v' \in V : v \odot (v \oplus v') = v$$

3.4.3 Similarity Function Formal Constraints

According to the Tversky's monotonicity axiom, similarity grows with the amount of common features (intersection) and decreases with the amount of disjoint features. Therefore, given the definition of π_V, its correspondence with π_S across π, and the definition of $\odot$ and $\oplus$, the similarity function should satisfying that the similarity between two vectors should be lower than the similarity between a vector and their specification (avoiding disjoint elements). On the other hand, the similarity between two vectors is higher than the similarity between a vector and its generalization (increasing the intersection size). That is,

Constraint 10 *[Similarity monotonicity]*

$$\delta(v_1, v_2) \leq \delta(v_1, v_1 \odot v_2), \quad \delta(v_1, v_2) \leq \delta(v_1, v_1 \oplus v_2)$$

In addition, specific meanings are closer to them self than general concepts:

Constraint 11 *[Self-similarity]*

$$\mathtt{I}(v_1) \geq \mathtt{I}(v_2) \longrightarrow \delta(v_1, v_1) \geq \delta(v_2, v_2)$$

The specific meaning is closer to the general than vice-versa. Formally,

Constraint 12 *[Asymmetricity] Being $v_1 \neq v_2$:*

$$\delta(v_1, v_1 \odot v_2) \geq \delta(v_1 \odot v_2, v_1)$$

4 Benchmarking representation systems

This section focuses in the desing of a test set to benchmark representation systems using word sequences, ontological ancestors and text summaries. First we analyze the similarities and differences of these three groups (sequences, ontological ancestors and summaries), then, based on this, we introduce a set of test for word sequences, ontological ancestors and summaries to check their compliance with the constraints defined in the previous section.

4.1 Word sequences, ontological ancestors and summaries

We first introduce some clarifications about working with word sequences, ontological ancestors and abstracts.

Word sequences: Following Constraints 2 and 3, adding more words to a sequence will increase the amount of information, as it is equivalent to over-specify a set of meanings to a subset of meanings.

Ontological ancestors: A word A is an ontological ancestor of a word B if there exists a hierarchical link between A and B, where A alludes to a more general or abstract concepts that B also posses. For example, the word 'dog' is more specific than its ontological ancestor 'mammal', and the word 'animal' is the ontological ancestor of both ’dog’ and ’mammal’. A word will be more specific than its ontological ancestor, and hence have more amount of than its ancestor.

Summaries: Less words in a sequence will contain less amount of information; hence, a summary must have less amount of information than the original text, as it is supposed to have a reduced extension compared with the original text. Furthermore, there exists a metric to measure the quality of a summary based on the information loss of the summary with respect of the original text (Hoplaros et al., 2014).

An extractive summary does not constitute a problem, as the original text will consist of the same words as the summary plus another set of words, so the above property will always be maintained. However, abstractive summaries may use some words that contain more information than the original words used in the text, so the summary will have more information than the original text, which does not make sense. If this happens, some extra information is being added, either deliberately (enriching it with one’s own knowledge, for example) or unconsciously (using a more formal register, for example). However, this is not a major problem, as it is expected to occur in very infrequent situations.

4.2 Test definition

The amount of information of word sequences, ancestors and summaries and the remarks stated in the previous paragraphs will allow us to design the same type of test (for some of the constraints) for all the three types of problem. Indeed, let s_1 and s_2 be two word sequences, and $s_1 + s_2$ be their concatenation. Then, we can assume that, in general,

$$\mathtt{I}(\pi(s_1)) \leq \mathtt{I}(\pi(s_1 + s_2)),$$

that is, the greater the number of words added to a sequence, the more information is transmitted. On the other hand, the amount of information contained in a word and its ontological ancestor has a similar relationship. The higher a word is in the ontological hierarchy, the more general it is, and therefore, it contains less information. So, let w be a word and w' be its ontological ancestor, then

$$\mathtt{I}(\pi(w')) \leq \mathtt{I}(\pi(w)).$$

A similar thing happens with summaries. A summary is a reduced representation of a text, so being s' a summary of s, then

$$\mathtt{I}(\pi(s')) < \mathtt{I}(\pi(s)).$$

Then, it is posible to design a test set to check if a Representation System complies with the Constraints from Section 3. Based on the similarities stated in the previous subsection, it is possible to design some of the tests that apply for word sequences, ancestors and summaries; others are different depending on the type of the test. In order to create the tests, we define:

- Being s_1, s_2, s_3 three word sequences with $\mathtt{I}(\pi(s_1)) > \mathtt{I}(\pi(s_2))$ and $s_1 + s_2$ the concatenation of the first two; s_3 another word sequence to use as support for comparison reasons.

- Being w_a an ancestor of both w and w' in an ontology (e.g. *mammal*, *cat* and *dog*), w_{aa} an ancestor of w_a (e.g. *animal* and *mammal*) and w_b having no relation with w_a.

- Being r_1 and r_2 summaries of r.

4.2.1 Tests for word sequences, ancestors and summaries

As stated before, some tests can be designed to apply for word sequences, summaries and ancestors. Then, we identify $s_1 = w = r_1$ and $s_2 = w_a = r_2$:

Test 1

$$\pi(s_1) \neq \pi(s_2)$$

Two different word sequences have two different representations.

Test 2

$$\mathtt{I}(\pi(""))=0$$

An empty sequence is represented as the zero vector.

Test 3

$$\pi(s_1)\odot\pi(s_2)=\pi(s_2)\odot\pi(s_1)$$

The specification function is conmutative.

Test 4

$$\pi(s_1)\oplus\pi(s_2)=\pi(s_2)\oplus\pi(s_1)$$

The generalization function is conmutative.

Test 5

$$(\pi(s_1)\odot\pi(s_2))\odot\pi(s_3)=\pi(s_1)\odot(\pi(s_2)\odot\pi(s_3))$$

The specification function is associative.

Test 6

$$(\pi(s_1)\oplus\pi(s_2))\oplus\pi(s_3)=\pi(s_1)\oplus(\pi(s_2)\oplus\pi(s_3))$$

The generalization function is associative.

Test 7

$$(\pi(s_1) \oplus \pi(s_2)) \odot \pi(s_3) = (\pi(s_1) \odot \pi(s_3)) \oplus (\pi(s_2) \odot \pi(s_3))$$

The generalization function is distributive.

Test 8

$$(\pi(s_1) \odot \pi(s_2)) \oplus \pi(s_3) = (\pi(s_1) \oplus \pi(s_3)) \odot (\pi(s_2) \oplus \pi(s_3))$$

The specification function is distributive.

Test 9

$$\delta[(\pi(s_1)\oplus\pi(s_2))\oplus\pi(s_1), \pi(s_1)\oplus\pi(s_2)] > \delta[(\pi(s_1)\oplus\pi(s_2))\oplus\pi(s_1), (\pi(s_1)\oplus\pi(s_2))\oplus\pi(s_3)]$$

To check idempotency of the generalization function: $(\pi(s_1) \oplus \pi(s_2)) \oplus \pi(s_1)$ *must be more similar to* $\pi(s_1) \oplus \pi(s_2)$ *(ideally, it must be equal) than for any other vector.*

Test 10

$$\delta[(\pi(s_1)\odot\pi(s_2))\odot\pi(s_2),\pi(s_1)\odot\pi(s_2)] > \delta[(\pi(s_1)\odot\pi(s_2))\odot\pi(s_2),(\pi(s_1)\odot\pi(s_2))\odot\pi(s_3)]$$

To check idempotency of the specification function: $(\pi(s_1)\odot\pi(s_2))\odot\pi(s_2)$ *must be more similar to* $\pi(s_1)\odot\pi(s_2)$ *(ideally, it must be equal) than for any other vector.*

Test 11

$$\delta[(\pi(s_1)\oplus\pi(s_2))\odot\pi(s_1),\pi(s_1)] > \delta[(\pi(s_1)\oplus\pi(s_2))\odot\pi(s_1),\pi(s_2)]$$

To check neutral elements: specifying over something that has been generalized must result in the original concept, then, $(\pi(s_1)\oplus\pi(s_2))\odot\pi(s_1)$ *must be more similar to* $\pi(s_1)$ *(ideally, it must be equal) than for any other vector in the space.*

Test 12

$$\delta[(\pi(s_1)\odot\pi(s_2))\oplus\pi(s_1),\pi(s_1)] > \delta[(\pi(s_1)\odot\pi(s_2))\oplus\pi(s_1),\pi(s_2)]$$

To check neutral elements: generalizing over something that has been specified must result in the original concept, then, $(\pi(s_1)\odot\pi(s_2))\oplus\pi(s_1)$ *must be more similar to* $\pi(s_1)$ *(ideally, it must be equal) than for any other vector in the space.*

Test 13

$$\delta(\pi(s_1), \pi(s_1) \odot \pi(s_2)) > \delta(\pi(s_1), \pi(s_2))$$

To check similarity monotonocity: a concept A is less similar to another concept B than to the specification of A with B.

Test 14

$$\delta(\pi(s_1), \pi(s_1) \oplus \pi(s_2)) > \delta(\pi(s_1), \pi(s_2))$$

To check similarity monotonocity: a concept A is more similar to another concept B than to the generalization of A with B.

Test 15

$$\delta(\pi(s_1), \pi(s_1)) > \delta(\pi(s_2), \pi(s_2))$$

To check self-similarity: the more specific is the sequence, the more similar is to itself.

Test 16

$$\delta(\pi(s_1), \pi(s_1) \odot \pi(s_2)) > \delta(\pi(s_1) \odot \pi(s_2), \pi(s_1))$$

To check that a specific meaning is closer to the general than vice versa.

Test 17

$$\delta(\pi(s_1), \pi(s_1) \oplus \pi(s_2)) < \delta(\pi(s_2), \pi(s_1) \oplus \pi(s_2))$$

To check that a specific meaning is closer to the general than vice versa.

4.2.2 Tests using word sequences

These tests only apply for the word sequences dataset.

Test 18

$$\mathtt{I}(\pi(s_1)) < \mathtt{I}(\pi(s_1 + s_2))$$

Over-specifying implies more amount of information.

Test 19

$$\mathtt{I}(\pi(s_1)), \mathtt{I}(\pi(s_2)) < \mathtt{I}(\pi(s_1) \odot \pi(s_2))$$

The specification has more amount of information than the components alone.

Test 20

$$\mathtt{I}(\pi(s_1) \oplus \pi(s_2)) < \mathtt{I}(\pi(s_1)), \mathtt{I}(\pi(s_2))$$

The generalization has less amount of information than the components alone.

4.2.3 Tests using ontological ancestors

These tests only apply for the ontological ancestors dataset.

Test 21

$$\mathtt{I}(\pi(w_a)) \neq \mathtt{I}(\pi(w))$$

A word can not have the same amount of information than its ontological ancestor (as it is more specific).

Supposing that $\delta(w, w_a) > \delta(w, w_b)$, then

Test 22

$$\mathtt{I}(\pi(w) \oplus \pi(w_a)) > \mathtt{I}(\pi(w) \oplus \pi(w_b))$$

The amount of information of generalizing two similar concepts is more than generalizing two different concepts.

Test 23

$$\mathtt{I}(\pi(w) \odot \pi(w_a)) < \mathtt{I}(\pi(w) \odot \pi(w_b))$$

The amount of information of specifying two similar concepts is less than specifying two different concepts.

4.2.4 Tests using summaries

These tests only apply for the summaries dataset.

Test 24

$$\mathtt{I}(\pi(r)) > \mathtt{I}(\pi(r_1))$$

A summary contain less information than the original text.

Test 25

$$\mathtt{I}(\pi(r_1)), \mathtt{I}(\pi(r_2)) < \mathtt{I}(\pi(r_1) \odot \pi(r_2))$$

The specification has more amount of information than the components alone.

Test 26

$$\mathtt{I}(\pi(r_1) \oplus \pi(r_2)) < \mathtt{I}(\pi(r_1)), \mathtt{I}(\pi(r_2))$$

The generalization has less amount of information than the components alone.

Table 1 shows the tests for words sequences, ancestors and summaries associated to each of the defined constraints.

4.3 Dataset definiton

Once the tests have been defined, we have proceeded to create the dataset with which to run the tests. This dataset consists on 70 word sequences, 47 ancestors and 40

<table>
<tr><th>Constraint</th><th>Tests with sequences</th><th>Tests with ancestors</th><th>Tests with summaries</th></tr>
<tr><td>Constraint 1</td><td colspan="3">Test 1</td></tr>
<tr><td>Constraint 2</td><td colspan="3">Test 2</td></tr>
<tr><td>Constraint 3</td><td>Test 18</td><td>Test 21</td><td>Test 24</td></tr>
<tr><td>Constraint 4</td><td>Test 19, 20</td><td>Test 22, 23</td><td>Test 25, 26</td></tr>
<tr><td>Constraint 5</td><td colspan="3">Test 3, 4</td></tr>
<tr><td>Constraint 6</td><td colspan="3">Test 5, 6</td></tr>
<tr><td>Constraint 7</td><td colspan="3">Test 7, 8</td></tr>
<tr><td>Constraint 8</td><td colspan="3">Test 9, 10</td></tr>
<tr><td>Constraint 9</td><td colspan="3">Test 11, 12</td></tr>
<tr><td>Constraint 10</td><td colspan="3">Test 13, 14</td></tr>
<tr><td>Constraint 11</td><td colspan="3">Test 15</td></tr>
<tr><td>Constraint 12</td><td colspan="3">Test 16, 17</td></tr>
</table>

Table 1: Tests for words sequences, ancestors and summaries associated to each of the defined constraints.

summaries, thus creating 157 elements, in order to test different scenarios and to be able to analyse the impact on the type of data on the robustness of the framework.

A first proposal with the test cases was created by ChatGPT and then all test cases proposed were finally selected manually, as several cases proposed from ChatGPT, specially for the ancestors dataset, were difficult to understand the hierarchical relations among the words.

The results are aggregated and normalized so each of the constraints and the three type of data have a score from 0 to 1, making then a 36 point grading system (12 points for each type: sequences, ancestors and summaries). Some examples of the dataset can be found in Tables 2 (for sequences data), 3 (for ancestors data) and 4 (for summaries data).

s_2	s_1	s_3
dog	blue car	apple pie is delicious
cat	tall tree	running in the park
hat	red balloon	playing guitar solo
house	green grass	swimming in the lake
mouse	big elephant	dancing under the moon

Table 2: Example for the sequences dataset.

w	w'	w_a	w_{aa}
Dog	Cat	Mammal	Animal
Pear	Apple	Fruit	Plant
City	Town	Settlement	Place
Guitar	Drums	Instrument	Music
Ash	Tree	Forest	Plant

Table 3: Example for the ancestors dataset.

r	r_1	r_2
The quick brown fox jumps over the lazy dog.	A fox jumps over a dog.	Animal jumps over animal.
John is a talented musician who plays multiple instruments.	John is a talented musician.	John plays several instruments
The majestic mountains stood tall, covered in a blanket of snow.	Majestic snow-covered mountains.	Snowy mountains in their glory.
Emily spent her summer vacation traveling to exotic locations and exploring new cultures.	Emily explored new cultures on summer vacation.	Traveler Emily discovers exotic places.
The scientific experiment yielded groundbreaking results that could revolutionize the industry.	Groundbreaking experiment results.	Revolutionary findings in science.

Table 4: Example for the summaries dataset.

5 Experimentation

In this section we define several representation systems (i.e., several embedding, information measurement, generalization, specification and similarity functions) to evaluate the framework defined in the previous section, and then we present and analyze the results.

5.1 Definition of the representation systems

In order to test our evaluation framework, we applied it to a series of approaches to embedding functions, information measurement, generalization, specification, and similarity.

5.1.1 Information Measurement function

For the information Measurement function, we consider the vector's norm. According to the analysis by Levy and Goldberg (2014) and Arora et al. (2016), the dot product of SGNS embedding (Skip-gram with Negative-Sampling) approximates the Pointwise Mutual Information (PMI) between two words. With $\pi(w)$ being the embedding of the word w:

$$\langle \pi(w), \pi(w') \rangle \propto \mathrm{PMI}(w, w') = \log \left(\frac{P(w, w')}{P(w') \cdot P(w')} \right)$$

We will refer to this as the Levy's correspondence. This implies that there exists a correspondence between the vector norm and the Information Content of represented utterances according to Shannon's Information Theory:

$$\mathrm{IC}(w) = -\log(P(w)) = -\log\left(\frac{P(w,w)}{P(w)\cdot P(w)}\right) = \mathrm{PMI}(w,w) \simeq \langle\pi(w),\pi(w)\rangle = \|\pi(w)\|^2$$

The information measurement function consists in extending this correspondence to word sequences:

$$I(v) = |v|^2$$

In addition, Gao et al. (2019b) proved that under some assumptions, the optimal embeddings of infrequent tokens in Transformer Language Models can be extremely far away from the origin. Li et al. (2020) observed empirically that *"high-frequency words are all close to the origin, while low-frequency words are far away from the origin"* in Transformer language models.

5.1.2 Embedding Functions

As embedding function, we use in this experiment the SGNS implementation Word2Vec, which is the most popular word embedding function in the literature. In order to represent word sequences, we apply five alternative word composition functions:

’Sum’ and ’average’ word composition functions: The first two are the sum and average of vectors. A large body of literature has shown that the sum or global average of word embeddings is very effective, often outperforming more sophisticated methods (Mitchell and Lapata, 2010; Boleda, 2020; Lenci, 2018; Blacoe and Lapata, 2012; Perone et al., 2018; Baroni and Lenci, 2010; Rimell et al., 2016; Czarnowska et al., 2019; Wieting and Gimpel, 2018; Ethayarajh, 2018).

Information Theory based word composition functions (F$_{joint}$, F$_{ind}$ and F$_{inf}$): An intrinsic limitation of these two additive approaches is that word order is not considered. In Amigó et al. (2022) several Information Theoretic functions that preserve the text structure are presented. These composition functions particularise the $F_{\lambda,\mu}(v_1, v_2)$ function for different λ and μ values:

$$F_{\lambda,\mu}(v_1, v_2) = \frac{v_1 + v_2}{|v_1 + v_2|}\sqrt{\lambda(|v_1|^2 + |v_2|^2) - \mu(v_1 \cdot v_2)},$$

The F$_{joint}$ variant (with $\lambda = 1$ and $\mu = 1$) assumes that the information content of the composition of two sequences s_1 and s_2 corresponds with the joint probability ($I(s) = -log(P(s_1, s_2))$). The second variant F$_{ind}$, with $\lambda = 1, \mu = 0$, assumes that statistical independence between words ($I(s) = -log(P(s_1) \cdot P(s_2))$). The third variant, F$_{inf}$, with $\lambda = 1, \mu = \frac{min(|v_1|, |v_2|)}{max(|v_1|, |v_2|)}$, is designed to satisfy different formal constraints (Amigó et al., 2022).

5.1.3 Specification and generalization functions

We define three pairs as specification and generalization functions.

Algebraic approach (Algebraic): In terms of generalizing two concepts, the idea of algebraic basis has been used. Every vector $v \in V$ can be written as a weighted sum of the vectors that compounds the basis. Then, a vector (a subset of meanings) can be seen as a composition of different vectors (subsets of meanings). To obtain the contribution of a specific vector w of the basis to the vector v, then it is needed to calculate the projection of v over w. This idea can be applied to two random vectors to get the contribution of one into the other and vice versa. This contribution can be seen as the generalization of the two vectors:

$$v_1 \oplus_{alg} v_2 = \frac{1}{2}\left[v_1 \cdot v_2 \frac{v_1}{|v_1|^2} + v_1 \cdot v_2 \frac{v_2}{|v_2|^2} \right]$$

Following these reasoning, the sum of vectors seems suitable for the specification functions, as the contributors of each of the vectors will increase its 'importance' if they are similar (then, there is an over-specification of this contributor) and decrease if they are opposite:

$$v_1 \odot_{alg} v_2 = v_1 + v_2$$

Their limitation is the associativity (for generalization) and distributiveness (for both generalization and specification) properties, which are not satisfied.

Information Theory based functions (Information): As a second option, we start from the Levy's correspondence (PMI vs. dot product of representations), extending this principle from words to sequences. Being $v_1 = \pi(s_1)$ and $v_2 = \pi(s_2)$:

$$\mathrm{PMI}(s_1, s_2) \propto v_1 \cdot v_2$$

Therefore, the mutual information between two sequences can be estimated as the dot product multiplied by a certain factor β. For instance, *"pets"* could be considered as the mutual information of *"dogs"* and *"cats"*. We define the generalization function in each dimension as the common information of sequences:

$$(v_1 \oplus_{inf} v_2)_i = \mathrm{MinNormSign}(v_{1,i}, v_{2,i})\sqrt{\beta \cdot v_{1,i} \cdot v_{2,i}}$$

where $\mathrm{MinNormSign}(v_1, v_2)$ represents the sign of the lower component between $\{v_{1,i}, v_{2,i}\}$. The sign selection gives preference to the most general meaning. On the other hand, the information based specification is defined as the sum of Information Quantities minus the mutual (common redundant) information, giving preference to the longest vector

(most specific meaning)

$$(v_1 \odot_{inf} v_2)_i = \text{MaxNormSign}(v_1, v_2)\sqrt{v_{1,i}^2 + v_{2,i}^2 + \beta \cdot v_{1,i} \cdot v_{2,i}}$$

where $\text{MaxNormSign}(v_1, v_2)$ represents the sign of the major component between $\{v_{1,i}, v_{2,i}\}$. Assuming the Levy's PMI correspondence, these functions satisfy that the Information Quantity of the combination is a linear function of the single and mutual Information Quantities:

$$\begin{aligned} I(v_1 \oplus_{inf} v_2) &= |v_1 \oplus_{inf} v_2|^2 = \sum_i (\beta v_{1,i} v_{2,i}) \\ &= \beta v_1 \cdot v_2 \simeq PMI(s_1, s_2) \\ I(v_1 \odot_{inf} v_2) &= |v_1 \odot_{inf} v_2|^2 = \sum_i (v_{1,i}^2 + v_{2,i}^2 - \beta v_{1,i} v_{2,i}) \\ &= v_1^2 + v_2^2 - \beta v_1 \cdot v_2 \simeq I(s_1) + I(s_2) - PMI(s_1, s_2) \end{aligned}$$

In addition, taking $\beta = \frac{min(|v_1|,|v_2|}{max(|v_1|,|v_2|)}$, these functions satisfy the boundary constraints:

$$I(v_1 \oplus_{inf} v_2) < I(v_1), I(v_2)$$

$$I(v_1 \odot_{inf} v_2) > I(v_1), I(v_2)$$

Their limitation is the associativity, distributiveness and idempotency properties, which are not satisfied.

Fuzzy Set Operator based functions (Fuzzy): In order to satisfy the associativity property, we apply the union and intersection operators to the vectors. The intersection of fuzzy sets (minimum value in each component) reduces the representation to common information, while the union operator (maximum value) generates more specific representations, but eliminates redundant information. We handle the sign in the same way as in the previous functions.

$$(v_1 \oplus_{SetOp} v_2)_i = \mathrm{MinNormSign}(v_1, v_2)\,\mathrm{Min}(|v_{1,i}|, |v_{2,i}|)$$

$$(v_1 \odot_{SetOp} v_2)_i = \mathrm{MaxNormSign}(v_1, v_2)\,\mathrm{Max}(|v_{1,i}|, |v_{2,i}|)$$

The limitation of the fuzzy set operator based functions is that they do not keep a direct correspondence with Information Quantities according to the Levy's correspondence.

5.1.4 Similarity Functions

As similarity functions we consider the alternatives proposed in Amigó et al. (2022). This work includes the standard similarity functions Cosine, Euclidean distance, and dot product, as well as the Information Contrast Model with is based on Information

Theory (Amigó et al., 2020):

$$
\begin{aligned}
\delta_{cos}(v_1, v_2) &= \frac{v_1 \cdot v_2}{|v_1||v_2|}, \\
\delta_{Euc}(v_1, v_2) &= \sum_{i=1}^{d} (v_1^i - v_2^i)^2, \\
\delta_{DotProd}(v_1, v_2) &= v_1 \cdot v_2, \\
\delta_{ICM}(v_1, v_2) &= \alpha_1 IC(v_1) + \alpha_2 IC(v_2) - \beta IC(v_1 \cup v_2).
\end{aligned}
$$

The information contrast model (Amigó et al., 2020) is the linear combination of their Information Quantities and the Information Quantity of their union:

$$\delta(v_1, v_2) = \alpha_1 IC(v_1) + \alpha_2 IC(v_2) - \beta IC(v_1 \cup v_2),$$

which can be seen as a generalized version of a parametric Pointwise Mutual Information. The ICM parameters were set to $\alpha_1 = 1 = \alpha_2$ and $\beta = 1.5$.

5.2 Results

Combining the corresponding embedding functions, generalization and specification functions and similarity functions we have $5 \times 3 \times 4 = 60$ combinations. An analysis of all these possible combination has been performed. For each of the combination, all tests were executed, obtaining a score for each of the constraints and the type of test (sequences, ancestors, summaries). The results of all combinations can be found in Tables 5 and 6. Each row show the results of one of the possible combinations, and each column show the result for a type of test (sequence, ancestors, summaries, and total). Values represent the score obtained throughout the execution, with a range from 0 to 36, being the higher, the better. In bold letter, the highest result for each of the column. The best results are obtained with the combination of Sum of Vectors, Fuzzy approach and ICM, although the $F_{\lambda,\mu}$ functions obtain a similar score as the Sum of Vectors. The results of the combinations will be studied deeper in the next subsections, from an embedding, specification and generalization and similarity perspective.

Note that the embedding functions do not apply to the ancestors tests results, i.e. the results are the same for the sum of vectors, average and F functions (they depend on the generalization and specification functions and similarity functions). This is due to the fact that the embedding functions are different when composing several words, and ancestor tests are single-word tests.

5.2.1 Embedding function analysis

Results are shown in Figure 7 and Table 7. $F_{\lambda,\mu}$ functions obtain a similar score to the Sum of vectors. Averaging the vectors is the worst choice. This is because the larger the text, the greater the amount of information, and the more impact this amount of information has on the rest of the processes. An embedding that does not 'eliminate'

Type	Sequences	Ancestors	Summaries	Total
Sum of Vectors - Algebraic - Cosine	8.86	8.11	6.82	23.79
Sum of Vectors - Algebraic - Euclidean	6.09	5.67	5.76	17.52
Sum of Vectors - Algebraic - Dot	7.74	7.66	8.77	24.17
Sum of Vectors - Algebraic - ICM	8.66	8.34	9.25	26.25
Sum of Vectors - Information - Cosine	8.69	7.93	6.85	23.47
Sum of Vectors - Information - Euclidean	5.25	4.52	5.01	14.78
Sum of Vectors - Information - Dot	8.42	8.14	8.80	25.36
Sum of Vectors - Information - ICM	8.64	8.13	9.45	26.22
Sum of Vectors - Fuzzy - Cosine	9.62	8.76	7.62	26.00
Sum of Vectors - Fuzzy - Euclidean	6.25	5.46	6.01	17.72
Sum of Vectors - Fuzzy - Dot	9.37	8.68	9.53	27.58
Sum of Vectors - Fuzzy - ICM	9.60	**9.11**	**10.49**	**29.20**
Average - Algebraic - Cosine	7.78	8.11	7.09	22.98
Average - Algebraic - Euclidean	5.01	5.67	5.45	16.13
Average - Algebraic - Dot	6.47	7.66	5.69	19.82
Average - Algebraic - ICM	7.53	8.34	6.09	21.96
Average - Information - Cosine	7.77	7.93	8.00	23.70
Average - Information - Euclidean	4.33	4.52	4.53	13.38
Average - Information - Dot	7.58	8.14	6.21	21.93
Average - Information - ICM	7.77	8.13	7.25	23.15
Average - Fuzzy - Cosine	8.77	8.76	8.61	26.14
Average - Fuzzy - Euclidean	5.33	5.46	5.53	16.32
Average - Fuzzy - Dot	8.48	8.68	7.11	24.27
Average - Fuzzy - ICM	8.75	**9.11**	8.14	26.00

Table 5: Results for all the combinations tested (Part 1/2). Each row shows the results of one of the possible combinations, and each column show the result for a type of test (sequence, ancestors, summaries, and total). Values represent the score obtained throughout the execution, with a range from 0 to 36, being the higher, the better. In bold letter, the highest result for each of the column. The best results are obtained with the combination of Sum of Vectors, Fuzzy approach and ICM, although the $F_{\lambda,\mu}$ functions obtain a similar score as the Sum of Vectors.

Type	Sequences	Ancestors	Summaries	Total
F$_{joint}$ - Algebraic - Cosine	8.43	8.11	7.34	23.88
F$_{joint}$ - Algebraic - Euclidean	5.58	5.67	5.70	16.95
F$_{joint}$ - Algebraic - Dot	7.12	7.66	8.00	22.78
F$_{joint}$ - Algebraic - ICM	8.11	8.34	8.97	25.42
F$_{joint}$ - Information - Cosine	8.77	7.93	8.31	25.01
F$_{joint}$ - Information - Euclidean	5.26	4.52	5.02	14.80
F$_{joint}$ - Information - Dot	8.50	8.14	8.94	25.58
F$_{joint}$ - Information - ICM	8.63	8.13	9.34	26.10
F$_{joint}$ - Fuzzy - Cosine	9.70	8.76	8.80	27.26
F$_{joint}$ - Fuzzy - Euclidean	6.26	5.46	6.01	17.73
F$_{joint}$ - Fuzzy - Dot	9.35	8.68	9.25	27.28
F$_{joint}$ - Fuzzy - ICM	9.62	**9.11**	10.46	29.19
F$_{ind}$ - Algebraic - Cosine	8.39	8.11	6.58	23.08
F$_{ind}$ - Algebraic - Euclidean	5.58	5.67	5.58	16.83
F$_{ind}$ - Algebraic - Dot	7.17	7.66	8.17	23.00
F$_{ind}$ - Algebraic - ICM	8.13	8.34	8.91	25.38
F$_{ind}$ - Information - Cosine	8.73	7.93	7.33	23.99
F$_{ind}$ - Information - Euclidean	5.25	4.52	4.99	14.76
F$_{ind}$ - Information - Dot	8.47	8.14	8.81	25.42
F$_{ind}$ - Information - ICM	8.64	8.13	9.37	26.14
F$_{ind}$ - Fuzzy - Cosine	9.66	8.76	8.19	26.61
F$_{ind}$ - Fuzzy - Euclidean	6.25	5.46	5.99	17.70
F$_{ind}$ - Fuzzy - Dot	9.35	8.68	9.35	27.38
F$_{ind}$ - Fuzzy - ICM	9.64	**9.11**	10.44	29.19
F$_{inf}$ - Algebraic - Cosine	8.43	8.11	6.76	23.30
F$_{inf}$ - Algebraic - Euclidean	5.59	5.67	5.62	16.88
F$_{inf}$ - Algebraic - Dot	7.13	7.66	8.09	22.88
F$_{inf}$ - Algebraic - ICM	8.08	8.34	8.97	25.39
F$_{inf}$ - Information - Cosine	8.77	7.93	7.75	24.45
F$_{inf}$ - Information - Euclidean	5.26	4.52	5.01	14.79
F$_{inf}$ - Information - Dot	8.46	8.14	8.88	25.48
F$_{inf}$ - Information - ICM	8.61	8.13	9.41	26.15
F$_{inf}$ - Fuzzy - Cosine	**9.71**	8.76	8.47	26.94
F$_{inf}$ - Fuzzy - Euclidean	6.26	5.46	6.01	17.73
F$_{inf}$ - Fuzzy - Dot	9.30	8.68	9.32	27.30
F$_{inf}$ - Fuzzy - ICM	9.60	**9.11**	10.48	29.19

Table 6: Results for all the combinations tested (Part 2/2). Each row shows the results of one of the possible combinations, and each column show the result for a type of test (sequence, ancestors, summaries, and total). Values represent the score obtained throughout the execution, with a range from 0 to 36, being the higher, the better. In bold letter, the highest result for each of the column. The best results are obtained with the combination of Sum of Vectors, Fuzzy approach and ICM, although the $F_{\lambda,\mu}$ functions obtain a similar score as the Sum of Vectors.

the amount of information (by weighting, for example) must therefore offer better results. Specifically, averaging vectors impacts directly in Constraint 3, especially for great amount of words (such as summaries). Cosine similarity function hides this problem as it normalizes the vectors, but the other similarity functions, that are norm-sensitive, underperform due to this.

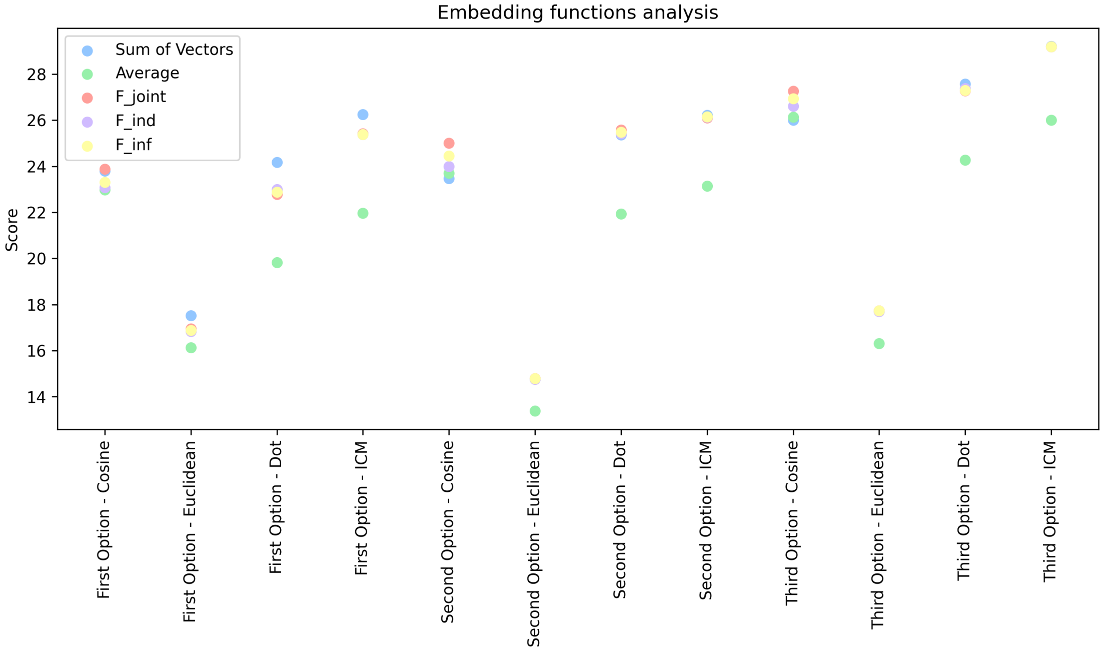

Figure 7: Visualization of the impact of the different embedding functions. X axis shows each of the combinations studied, Y axis shows the score obtained by each combination. Best results are obtained with the Fuzzy option + ICM, almost independently of the embedding function.

5.2.2 Specification and generalization functions analysis

Results are shown in Figure 8 and Table 8. The best results are obtained with the Fuzzy option, scoring around three points higher than the other two options and outperforming the two oher option in every combination. This is because it is the only associative

Type	Sum of vectors	Average	F$_{joint}$	F$_{ind}$	F$_{inf}$
Algebraic - Cosine	23.79	22.98	23.88	23.08	23.30
Algebraic - Euclidean	17.52	16.13	16.95	16.83	16.88
Algebraic - Dot	24.17	19.82	22.78	23.00	22.88
Algebraic - ICM	26.25	21.96	25.42	25.38	25.39
Information - Cosine	23.47	23.70	25.01	23.99	24.45
Information - Euclidean	14.78	13.38	14.80	14.76	14.79
Information - Dot	25.36	21.93	25.58	25.42	25.48
Information - ICM	26.22	23.15	26.10	26.14	26.15
Fuzzy - Cosine	26.00	**26.14**	27.26	26.61	26.94
Fuzzy - Euclidean	17.72	16.32	17.73	17.70	17.73
Fuzzy - Dot	27.58	24.27	27.28	27.38	27.30
Fuzzy - ICM	**29.20**	26.00	**29.19**	**29.19**	**29.19**

Table 7: Results grouped by embedding function. Each row shows the results of one of the possible combinations, and each column show the result for the embedding functions used. Values represent the score obtained throughout the execution, with a range from 0 to 36, being the higher, the better. In bold letter, the highest result for each of the column. Best results are obtained with the Fuzzy option + ICM, almost independently of the embedding function.

function of the three options, as the rest of the test results are similar.

5.2.3 Similarity function analysis

Results are shown in Figure 9 and Table 9. As it can be seen, the best results are obtained with ICM. Notice that ICM similarity measure generalizes the Ecludiean distance and dot product distance, but satisfying simoultaneosly more properties defined in Amigó et al. (2022). This is due to the fact that the ICM is an Information Theory measure: as it works with the amount of information of both components and their union, it is very sensitive with longer texts (summaries), even more if there is a lot of shared information between them.

Euclidean distance have the worst results (scoring around 6 points less than the

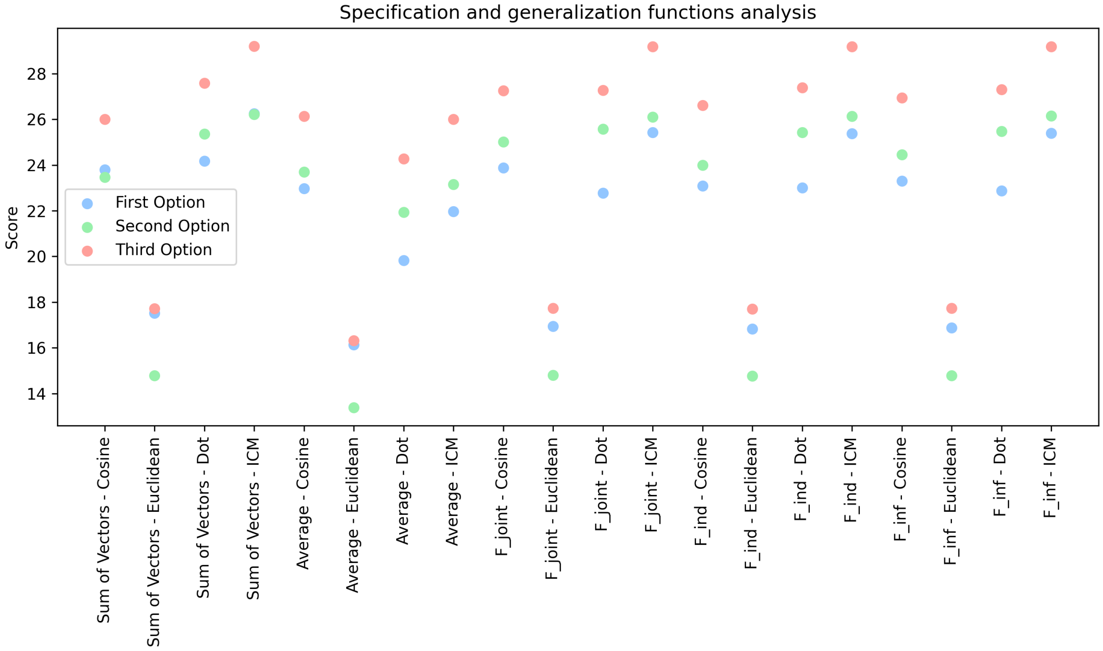

Figure 8: Visualization of the impact of the different specification and generalization functions. X axis shows each of the combinations studied, Y axis shows the score obtained by each combination. Best results are obtained with the Sum of Vectors + ICM, although $F_{\lambda,\mu}$ functions + ICM obtain similar results.

Type	Algebraic	Information	Fuzzy
Sum of Vectors - Cosine	23.79	23.47	26.00
Sum of Vectors - Euclidean	17.52	14.78	17.72
Sum of Vectors - Dot	24.17	25.36	27.58
Sum of Vectors - ICM	**26.25**	**26.22**	**29.20**
Average - Cosine	22.98	23.70	26.14
Average - Euclidean	16.13	13.38	16.32
Average - Dot	19.82	21.93	24.27
Average - ICM	21.96	23.15	26.00
F_{joint} - Cosine	23.88	25.01	27.26
F_{joint} - Euclidean	16.95	14.80	17.73
F_{joint} - Dot	22.78	25.58	27.28
F_{joint} - ICM	25.42	26.10	29.19
F_{ind} - Cosine	23.08	23.99	26.61
F_{ind} - Euclidean	16.83	14.76	17.70
F_{ind} - Dot	23.00	25.42	27.38
F_{ind} - ICM	25.38	26.14	29.19
F_{inf} - Cosine	23.30	24.45	26.94
F_{inf} - Euclidean	16.88	14.79	17.73
F_{inf} - Dot	22.88	25.48	27.30
F_{inf} - ICM	25.39	26.15	29.19

Table 8: Results grouped by specification and generalization functions. Each row shows the results of one of the possible combinations, and each column show the result for the specification and generalization functions used. Values represent the score obtained throughout the execution, with a range from 0 to 36, being the higher, the better. In bold letter, the highest result for each of the column. Best results are obtained with the Sum of Vectors + ICM, although $F_{\lambda,\mu}$ functions + ICM obtain similar results.

other similairy functions). This may be due to the fact that, at the distance level, a similarity measure generated by an angular difference (more semantic) is not the same as a modulus (amount of information). Hence, a measure that gives the same weight to both components cannot result in a good overall measure.

The dot product obtains slightly better results than the cosine. They are, essentially, the same metric, the cosine being normalized and the dot product not being normalized. It seems that some information is lost normalizing the results. As the module of the vectors is related to their Information Quantity, this means that Information Quantity is relevant in order to compute the similarity between vectors.

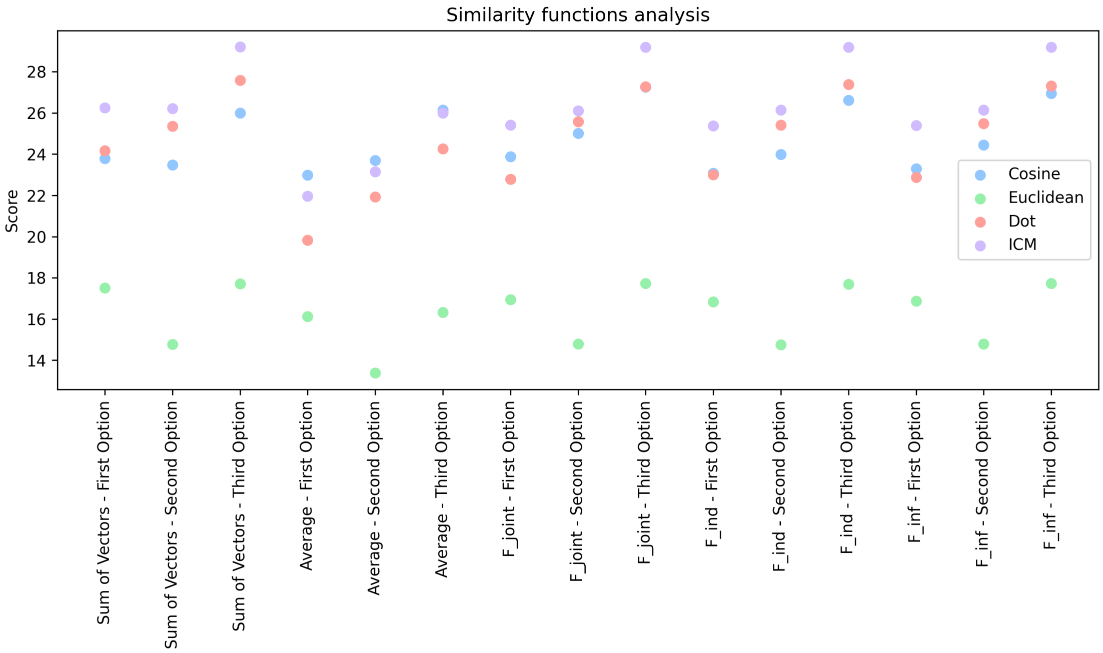

Figure 9: Visualization of the impact of the different similarity functions. X axis shows each of the combinations studied, Y axis shows the score obtained by each combination. Best results are obtained with the Fuzzy option, both for Sum of Vectors and $F_{\lambda,\mu}$ functions.

Type	Cosine	Euclidean	Dot	ICM
Sum of Vectors - Algebraic	23.79	17.52	24.17	26.25
Sum of Vectors - Information	23.47	14.78	25.36	26.22
Sum of Vectors - Fuzzy	26.00	17.72	**27.58**	**29.20**
Average - Algebraic	22.98	16.13	19.82	21.96
Average - Information	23.70	13.38	21.93	23.15
Average - Fuzzy	26.14	16.32	24.27	26.00
F$_{joint}$ - Algebraic	23.88	16.95	22.78	25.42
F$_{joint}$ - Information	25.01	14.80	25.58	26.10
F$_{joint}$ - Fuzzy	**27.26**	**17.73**	27.28	29.19
F$_{ind}$ - Algebraic	23.08	16.83	23.00	25.38
F$_{ind}$ - Information	23.99	14.76	25.42	26.14
F$_{ind}$ - Fuzzy	26.61	17.70	27.38	29.19
F$_{inf}$ - Algebraic	23.30	16.88	22.88	25.39
F$_{inf}$ - Information	24.45	14.79	25.48	26.15
F$_{inf}$ - Fuzzy	26.94	**17.73**	27.30	29.19

Table 9: Results grouped by similarity function. Each row shows the results of one of the possible combinations, and each column show the result for the similarity functions used. Values represent the score obtained throughout the execution, with a range from 0 to 36, being the higher, the better. In bold letter, the highest result for each of the column. Best results are obtained with the Fuzzy option, both for Sum of Vectors and $F_{\lambda,\mu}$ functions.

6 Conclusions and futures lines of work

6.1 Conclusions

We have shown that Information Theory based functions performs as well as the most used functions, even outperforming them: F-functions and the vector sum obtain similar results when considering the trade-offs with the rest of the functions involved, while vector averaging obtains inferior results when 'smoothing' the information quantity; and we have also shown that the ICM is the best measure of similarity from a holistic point of view, considering the trade-offs with the rest of the functions involved.

Additionally, we have stressed the necessity of similarity functions more sensitive to the amount of information: we have shown that, considering the trade-offs with the rest of the functions involved, the Euclidean distance is the worst performer, which demonstrates that the information quantity (modulus) and the semantics (angle) do not have the same impact on the result. This also explains why the dot product has a slightly better result than the cosine, since the cosine only considers the similarity by angle as it is a normalised result. Combining these two results, we can determine that the greatest impact on similarity is given by the angle, having the amount of information a little impact on actual similarity functions.

We have also shown that the presented framework is robust, and that results obtained testing several proposals such as the F-functions or ICM are consistent with the results obtained by their authors. This demonstrates that the goal of this work of creating a framework that can be used by the community as a reference for future research in representation systems has been achieved.

6.2 Future research

Several lines of research are opened with this work:

- The main research focus will be on conducting a comparative study of the main current proposals, whether contextual or non-contextual, in order to establish a classification of the best possible combinations to be used.
- A second line of work will focus on the creation of new properties resulting from the transfer of the rules of the set algebra operators, in order to enrich the benchmark presented in this work.
- A third line of work will focus on the search for new functions of the representation system: new specification and generalization functions, similarity functions that are more sensitive to the information quantity, for example.
- A fourth line of work should focus on continuing to bridge the gap between distributional semantics and information theory.

Bibliography

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Enrique Amigó, Fernando Giner, Julio Gonzalo, and Felisa Verdejo. On the foundations of similarity in information access. *Information Retrieval Journal*, 23(3):216–254, 2020.

Enrique Amigó, Alejandro Ariza-Casabona, Victor Fresno, and M Antònia Martí. Information theory–based compositional distributional semantics. *Computational Linguistics*, 48(4):907–948, 2022.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*, 2017.

Alejandro Fuster Baggetto and Víctor Fresno. Is anisotropy really the cause of bert embeddings not being semantic? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4271–4281, 2022.

Raphael Bailly and Laurent Leblond. Syntax and geometry of information. In *61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Marco Baroni. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B*, 375(1791):20190307, 2020.

Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

Jasmijn Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. Jump to better conclusions: Scan both left and right. *arXiv preprint arXiv:1809.04640*, 2018.

Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556, July 2012. URL https://www.aclweb.org/anthology/D12-1050.

Gemma Boleda. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234, jan 2020. doi: 10.1146/annurev-linguistics-011619-030303. URL `https://doi.org/10.1146%2Fannurev-linguistics-011619-030303`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nuo Chen, Linjun Shou, Ming Gong, Jian Pei, Bowen Cao, Jianhui Chang, Daxin Jiang, and Jia Li. Alleviating over-smoothing for unsupervised sentence representation. *arXiv preprint arXiv:2305.06154*, 2023.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryxjnREFwH`.

Stephen Clark and Stephen Pulman. Combining symbolic and distributional models of meaning. 2007.

Daoud Clarke, David Weir, and Rudi Lutz. Algebraic approaches to compositional distributional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, 2011.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.

Paula Czarnowska, Guy Emerson, and Ann Copestake. Words are vectors, dependencies are matrices: Learning word embeddings from dependency graphs. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 91–102, May 2019. doi: 10.18653/v1/W19-0408. URL https://www.aclweb.org/anthology/W19-0408.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Kawin Ethayarajh. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-3012. URL https://www.aclweb.org/anthology/W18-3012.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019a.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *ICLR*, 2019b. URL https://openreview.net/forum?id=SkEYojRqtm.

Mario Giulianelli, Jacqueline Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. *arXiv preprint arXiv:1808.08079*, 2018.

Google. An overview of bard: an early experiment with generative ai, March 2023. https://ai.google/static/documents/google-about-bard.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Demetris Hoplaros, Zahir Tari, and Ibrahim Khalil. Data summarization for network traffic monitoring. *Journal of network and computer applications*, 37:194–205, 2014.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Aishwarya Kamath and Rajarshi Das. A survey on semantic parsing. *CoRR*, abs/1812.00978, 2018. URL `http://arxiv.org/abs/1812.00978`.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

Alessandro Lenci. Distributional models of word meaning. *Annual Review of Linguistics*, 4(1):151–171, 2018.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems 27*, pages 2177–2185, 2014.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *2020 Conference on EMNLP*, pages 9119–9130, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.733. URL `https://aclanthology.org/2020.emnlp-main.733`.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

Yoshihiro Maruyama. Compositionality and contextuality: the symbolic and statistical theories of meaning. In *Modeling and Using Context: 11th International and Interdisciplinary Conference, CONTEXT 2019, Trento, Italy, November 20–22, 2019, Proceedings 11*, pages 161–174. Springer, 2019.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017.

Meta. Llama-2: Open foundation and fine-tuned chat models, July 2023. https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b.

Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244, 2008.

Jeff Mitchell and Mirella Lapata. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

OpenAI. Introducing chatgpt, November 2022. https://openai.com/blog/chatgpt.

Christian S. Perone, Roberto Silveira, and Thomas S. Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *CoRR*, abs/1806.06259, 2018. URL `http://arxiv.org/abs/1806.06259`.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://aclanthology.org/N18-1202`.

Tamara Polajnar, Laura Rimell, and Stephen Clark. Evaluation of simple distributional compositional operations on longer texts. In *LREC*, pages 4440–4443, 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Laura Rimell, Jean Maillard, Tamara Polajnar, and Stephen Clark. RELPRON: A relative clause evaluation data set for compositional distributional semantics. *Computational Linguistics*, 42(4):661–701, December 2016. doi: 10.1162/COLI_a_00263. URL `https://www.aclweb.org/anthology/J16-4004`.

Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen Lee, and James T Kwok. Revisiting over-smoothing in bert from the perspective of graph. *arXiv preprint arXiv:2202.08625*, 2022.

P Smolensky, M Lee, X He, W-t Yih, J Gao, and L Deng. Basic reasoning with tensor product representations. arxiv, cs, 2016.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

Josef Valvoda, Naomi Saphra, Jonathan Rawski, Adina Williams, and Ryan Cotterell. Benchmarking compositionality with formal languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6007–6018, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Noortje J. Venhuizen, Petra Hendriks, Matthew W. Crocker, and Harm Brouwer. A framework for distributional formal semantics. In Rosalie Iemhoff, Michael Moortgat, and Ruy de Queiroz, editors, *Logic, Language, Information, and Computation*, pages 633–646, Berlin, Heidelberg, 2019. Springer Berlin Heidelberg. ISBN 978-3-662-59533-6.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3): 1–34, 2020.

John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th ACL (Vol. 1: Long Papers)*, pages 451–462, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1042. URL https://www.aclweb.org/anthology/P18-1042.

L.A Zadeh. Probability measures of fuzzy events. *Journal of Mathematical Analysis and Applications*, 23(2):421–427, 1968. ISSN 0022-247X. doi: https://doi.org/10.1016/0022-247X(68)90078-4. URL https://www.sciencedirect.com/science/article/pii/0022247X68900784.