UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA

Dealing with Language Emergent Behavior Using Vectors of Reduced Dimension

Luís António Rodrigues Gaspar Cordeiro

Mestrado Integrado em Engenharia Física

Dissertação orientada por:
João Carlos Caetano de Freitas Pires da Cruz
Hygor Piaget Monteiro Melo

2021

All language is a set of symbols
whose use among its speakers
assumes a share past

JORGE LUIS BORGES, THE ALEPH

Dedicated to my grandfather Luís

Acknowledgement

Numerous people were important in the effort that involved this thesis and they all deserve my deepest gratitude.

First of all, my advisors João Pires da Cruz and Hygor Piaget Melo with whom I had the pleasure of working and learning for the past year. To João for taking me as a student and suggesting such an interesting and out-of-the-box project. His wealth of knowledge and unique ways of thinking are an inspiration to me. And to Hygor for his precious feedback, help with the manuscript and for always providing advice that I highly value. I am very grateful to both.

To the teachers I came across in these years of academic formation and a special recognition to professors Olinda Conde and Guiomar Evans for their work as coordinators of the BSc and MSc in Physics Engineering at FCUL.

To my friends, to whom I have nothing but an immense gratitude for the support, for always making me laugh and for never taking me too seriously. In particular to *that squad* of people that accompanied me over the last 5 years and whose importance was far greater than anything I can put into words. It was quite a journey and I am blessed to have shared it with them.

Lastly, I thank from the bottom of my heart to my incredible family. Pai, Mãe, Mana, your unconditional love makes everything possible.

Abstract

Natural languages, as typical complex systems, exhibit distinctive properties arising from the relationships between their elements such as nonlinearity and emergence. Such properties, together with the high dimensionality inherent of extensive vocabularies, make natural languages intrinsically difficult to model. Word embedding models have tackled these difficulties by using distributional semantics along with neural-based models for computing vector representation of words in a space of reduced dimension. In particular, the word2vec model makes use of a 3-layer neural network that generates a vector space, Γ, where a quantitative notion of meaning is recovered. In this work, we use the word2vec architecture to show that, in the space of reduced dimension, in addition to meaning, it is also possible to recover a notion of word attractiveness. In this framework, we define in Γ the quantity mass, M, for each of the V words that form the vocabulary. It was found that M is positively correlated with the word frequencies in the text, f, and that both f and M are distributed according to power laws. It was also found that when the text is shuffled, that is, keeping word frequencies but changing their order, practically all words have $M = 0$ which suggests that mass is a property that does not bypass text's emergent structure. In addition, we have extended the definition of mass to serve as connection criterion for a new linguistic network (a model for languages in terms of a graph structure). It was found that this network exhibits scale-free and small-world properties and that its topology is significantly affected by text shuffling, on contrast to what is observed for other unsupervised linguistic networks. We also suggest that the total mass of the system may function as a measure that represents an intuitive concept of information and that is uniquely defined.

Keywords: Quantitative Linguistics, Complex Systems, Linguistic Networks, Word Embedding

Resumo da Tese

O surgimento das linguagens naturais é amplamente considerado como uma transição central na evolução da espécie humana. Criadas à margem de qualquer controlo centralizado, as linguagens naturais são objetos dinâmicos e em constante evolução cujas propriedades são frequentemente comparadas às de um típico sistema complexo. Tais objetos podem ser abordados enquanto sistemas compostos de unidades interdependentes - as palavras - que interagem e compõem um todo funcional com propriedades de não-linearidade, emergência e ordem-espontânea. O enquadramento das linguagens naturais no contexto dos sistemas complexos justifica, em parte, que as dificuldades inerentes à modelação destas linguagens contrastem quase paradoxalmente com a eficácia com que estas são processadas pelo cérebro humano.

Neste trabalho dedicam-se algumas secções a argumentar que modelos simplificados de linguagens naturais podem ser inadequados no sentido em que não têm em consideração o facto do texto ser primeiramente um vetor de informação caracterizado por ter uma estrutura complexa resultante das regras de sintaxe e semântica subjacentes à linguagem. Verificamos que tanto regularidades estatísticas (Lei de Zipf) como propriedades macroscópicas de redes linguísticas (modelos de linguagens em termos de estruturas tipo grafo) são praticamente inalteradas pelo simples processo de baralhamento aleatório da sequência das palavras numa amostra de texto. Tal sugere que as regularidades são consequência das frequências das palavras (uma vez que estas não são alteradas pelo processo de baralhamento) e não necessariamente evidências da estrutura emergente do texto.

A alta dimensionalidade do sistema linguístico surge como principal obstáculo a uma modelação na base de derivação direta de uma distribuição de probabilidades a partir de amostras de texto. Recentemente modelos de word embedding abordaram esta dificuldade ao representar as palavras num espaço de representação distribuída sob a forma de vetores reais. Em particular o modelo word2vec aplica as ideias da semântica distribucional utilizando uma arquitetura em rede neuronal pouco profunda que, quando treinada em quantidades substanciais de texto (não menos que 10^6 palavras), produz representações vetoriais que captam relações de ordem semântica. O algoritmo de word2vec é genérico e não supervisionado na medida em que é treinado a partir de texto não processado e não etiquetado.

Um dos objetivos deste trabalho passa por mostrar que no espaço vetorial gerado pelos vetores do word2vec, Γ, para além de uma noção de significado, é também possível recuperar uma noção de atratividade das palavras. Definiu-se em Γ uma métrica de atratividade, denominada por massa (*mass*), M, cujo valor é obtido com base nas duas matrizes de pesos do modelo word2vec, i.e., as matrizes que contém a informação que codifica todo o espaço vetorial Γ. Enquanto métrica de atratividade, pretende-se que a M reflita a versatilidade própria de cada palavra na

medida em que quanto mais versátil (ou genérico) o significado, mais atrativa esta será ao uso.

Treinou-se um modelo word2vec num texto com cerca de 14 milhões de palavras e avaliaram-se as massas individuais das 40 000 palavras mais frequentes. Verificou-se a existência de correlação positiva entre a massa M e a frequência f de cada uma das palavras. Observou-se também que M se distribui segundo leis de potência da forma:

$$P(M) \propto \frac{1}{M^{\beta}}$$

onde $P(M)$ é a probabilidade de uma palavra do vocabulário ter massa M e β é um expoente real positivo. A identificação de dois regimes em lei de potência na distribuição de $P(M)$ está em linha com observações relativas à distribuição de frequências e que foram anteriormente associadas há divisão do léxico em dois grupos: um léxico nuclear formado por um pequeno número de palavras versáteis e um léxico ilimitado para comunicação específica.

Submeteu-se ainda a definição de M a um teste relevante que consistiu em avaliar a massa de palavras num corpus aleatoriamente baralhado. Note-se que o baralhamento significa manter as frequências inalteradas, mas desfazer a estrutura do texto e perder o seu significado. Baralhando o mesmo texto com 14 milhões de palavras, verificou-se que mais de 99,9% das palavras registam massa $M = 0$. O facto da massa virtualmente não existir em texto baralhado indica que esta quantidade está diretamente ligada à estrutura emergente das linguagens naturais. As regras de sintaxe e semântica constituem regularidades que se traduzem em redundâncias no sistema texto. O processo de baralhamento elimina estas redundâncias e aumenta a imprevisibilidade da sequência de palavras. Consequentemente, o texto deixa de ser suficientemente regular para poder ser eficazmente representado no espaço de dimensão reduzida gerado pelo word2vec.

A definição de massa é estendida para servir como critério de ligação a uma nova rede linguística não-supervisionada apelidada de rede de embeddings (*embedding network*) cuja construção requer um texto que é primeiramente usado para treinar um modelo de word2vec. As arestas da rede de embeddings conectam palavras que tendem a coocorrer dentro da mesma janela de contexto. Observa-se que as palavras com maior conectividade correspondem a vocábulos com significados genéricos (*the*, *of*, *and*, *his*,) e que, na versão pesada da rede (onde cada aresta tem atribuído um peso real indicador da força da conexão), as ligações mais fortes estão associadas a combinações frequentes de unidades lexicais (*united+states* ou *hong+kong* por exemplo).

Em termos de propriedades macroscópicas, a rede de embeddings evidencia propriedades de rede livre de escala (*scale free*) e efeito de mundo pequeno (*small-world*). Uma vez que as conexões da rede de embeddings são baseadas na definição de M, observa-se que as propriedades da rede são significativamente afetadas pelo processo de baralhamento do texto. O mesmo não se verifica para outras redes linguísticas não supervisionadas onde o processo de baralhamento impacta as propriedades microscópicas, mas tem pouca reflexão a nível macroscópico.

Os resultados que indicam que M está intrinsecamente ligada à estrutura emergente da linguagem levam a que se considere a aplicabilidade desta métrica como medida de informação num sistema linguístico. No último capítulo de contribuições é realizada uma curta reflexão sobre a subjetividade dos conceitos de *informação* e *complexidade* no campo da teoria da informação onde são destacadas a Entropia de Shannon H_s e a Complexidade de Kolmogorov K

como principais definições quantitativas formais. Ambas as quantidades tendem a classificar como mais informativos sistemas com elevada aleatoriedade ou desordem. Acontece que, no contexto das linguagens naturais e em outros sistemas com propriedades emergentes, existe uma noção intuitiva de informação que implica mínimos para sistemas altamente regulares, bem como para sistemas altamente aleatórios.

A definição de M é novamente recuperada para servir como base a uma nova medida que representa uma noção mais intuitiva do termo "informação" no contexto das linguagens naturais. É definida a informação efetiva (*effective information*), Y, de um texto representado em Γ como o somatório das V massas individuais dos elementos do sistema:

$$Y = \sum_{i=0}^{V} M_i$$

.

A viabilidade de Y enquanto medida de informação assenta no cumprimento de três condições: (I) Em texto que não respeita as regras linguísticas de sintaxe/semântica (ou seja, texto sem significado), a informação efetiva deve ser nula; (II) Informação repetida ou redundante não deve contribuir para o aumento da informação efetiva; (III) Na ausência de irregularidades estruturais e redundâncias, a informação efetiva deve aumentar com o aumenta da quantidade de texto analisada.

Efetuaram-se testes com diferentes modelos de word2vec treinados e foi verificado que as condições (I) e (III) são respeitadas pela definição de Y. A verificação da condição (II) não foi conclusiva e sublinha-se que futuramente será importante testar de forma inequívoca a forma como Y é impactada por informação repetitiva ou redundante.

Palavras-Chave: Linguística Quantitativa, Sistemas Complexos, Redes Linguísticas, Vetorização Lexical

Content

List of Figures

Symbols and Acronyms

N	Embedding Space Dimensionality
V	Vocabulary Size
f	Frequency
c	Window Size
M_{w_1}	Mass of the word w_1
$\mathbf{w}_i^{M_1}$	Vector Representation of word i in matrix $\mathbf{M_1}$
Ω	Text Space
Γ	Embedding Space
S^{Δ}	Small World Coefficient
C	Clustering Coefficient
L	Average Shortest Path Length
k	Node Degree
e_i	Selectivity of node i
H	Entropy
H_s	Shannon's Entropy
K	Kolmogorov Complexity
Y	Effective Information
14MWC	14 Million Word Corpus
CBOW	Continuous Bag of Words
RBM	Restricted Boltzmann Machine
EC	Effective Complexity

Chapter 1

Introduction

This introductory chapter serves as a first exposure to the difficulties of modeling human languages due to the complexity of their emerging structure. It is also given a first outlook to the approaches and methods in the remaining chapters.

1.1 The Complexity of Natural Languages

Human language emergence is seen as a major transition in the evolution of the human species [1]. While the study of language origin's is not easy due to the lack of direct evidences, it is known that natural languages - language that evolved naturally in humans through use and repetition without a centralized planning - are complex and dynamic [2].

Natural languages allow the construction of a virtually infinite range of combinations from the limited set of words that form the vocabulary. The process of sentence generation in the brain is so efficient that using vocabularies composed by over 170 000 words - like the modern English one - seems like an effortless and instantaneous task [3].

Human's ease of use contrasts with the enormous difficulties that arise when trying to model a language to a computer interpretable way. It is not due to computational limitations that chat-bot communication is still so far behind real human communication when it comes to natural languages. Words are in fact complex. And if we start taking into account syntactic rules, ambiguous meaning words or neologisms, modeling a natural language becomes extremely challenging.

In fact, the organizational characteristics of language make it a topic that fits in complex systems theory [4]. Such systems are intrinsically difficult to model mostly due to the non explicit rules that make the system's organization and that lead to non explicit dependencies between the elements of that system [5, 6].

One key idea in complex systems, and therefore in text, is that the whole is not just the sum of the parts. The parts themselves carry little relevant information to the description of the system. It is in understanding the interactions between them that the information is found.

Using text one can produce clear examples of this just by considering three sentences:

1. Today Albert told me about his new theory of relativity.
2. New told of today relativity Albert me about theory his.
3. Relativity of theory new his about me told Albert today.

The reader certainly has no problem in identifying sentence 1 as *correct* and sentences 2 and 3 as *not so correct*. And since the three sentences are formed by the same ten words, it becomes obvious that the whole - *the informative content* - is not equal to the sum of the parts - *the words*. In system theory, this kind of behavior is called emergence [7].

While the study of complex systems is a well established field of science, there is still no unanimous criteria for identifying and measuring complexity [5]. In fact, that is probably why different authors address this with curious yet rather accurate analogy: "Complex Systems are like beauty: You know it when you see it" [8, 9].

Countless studies have addressed language in a statistical way and some finding unexpected regularities like Zipf's law, Heaps' law, Menzerath's law or Brevity law [10, 11, 12, 13, 14, 15, 16, 17]. Such regularities are seen as evidence of a particular type of emergency called self-organization, where forms of overall order arise from local interactions between parts without a

centralized control. In social sciences this phenom is often called *spontaneous order* and shortly defined as *"the result of human actions, not of human design"* [18].

Analysing text as just a sequence of words while forgetting that it is an information vector carrying a message from a sender to a receiver might be very tempting specially for anyone outside the context of social sciences and humanities. In this thesis we intend to show that this mindset of ignoring the emergent behaviour of linguistics can lead to vain results. Furthermore we propose neural-based approaches to insure that semantic and syntactic relations in text are not bypassed when modeling a linguistic system.

1.2 How does Physics relates with Language?

This work is framed within the field of study of complex systems which is an interdisciplinary domain consistently connected with the tools of statistical physics, complex network theory, nonlinear dynamics, computational science, linear algebra, information theory and others.

If we think of text as a system and words as the elements of that system, it is straightforward to guess that each element is unique and yet a concept of similarity between elements must exists. But that similarity has nothing to do with the characters that make a word. The property that actually differentiates words in our system is the semantic value, i.e., their meaning.

Furthermore, in such system the elements are not independent from each other. Which means they must be in some way correlated. To state this lets think of an actual text - a set of ordered words that, usually, form a set of ordered sentences; In every natural text there are signs of word correlation for example on collocations like *"Los Angeles Lakers"* where a set of words, when put together, has it's own semantic value. Statistically talking, words that form collocations or stereotyped expressions will have higher probabilities of co-occurring together.

A way to visualize the correlations between words is to think of the consequences of removing some of them: Studies show that the human brain can efficiently fill the blanks in text where some of the words were omitted [19, 20]. This sort of "autofill" function is due to the fact that word correlations result in redundancies that make text predictable up to a certain degree.

On larger scales, the same redundancies and correlations make it possible to quantify and categorize semantic similarities between linguistic units. This hypothesis, called *Distributional Hypothesis*, essentially states that words that tend to appear in similar contexts also have similar meanings. It is usually summarized by Firth's famous quote:

> *You shall know a word by the company it keeps.*
> J. R. Firth, 1957

Furthermore that implies that the information we get about the meaning of a word is acquired relatively to the surrounding words. This is a very important premise upon most of today's language modeling methods are based on and we will get back to it on Chapter 2 to explain this work's model.

A couple decades before Firth's distributional hypothesis, the Swiss linguist and philosopher

Ferdinand de Saussure developed and taught his approach of language as a formal system of differential elements. A self-contained, self-regulating system whose elements are defined by their relationship to other elements within the system [21]. In'the book *"Course in General Linguistics"*, Saussure gave special emphasis to the concept of *sign* formed by two distinct components: the *signified* - an idea or concept - and the *signifier* - the mean of expressing the signified (words for example). Although Saussure presented his views, which became known as structuralism and semiology, with formalisms from psychology and philosophy, the ideas remain relevant in quantitative linguistics and in the approach of languages as a complex system that would emerge almost a century after Saussure's death.

This work approach is to treat text as a highly heterogeneous and correlated system composed by a large number of elements - the words. And for that reason a prolific environment to apply physics's tools. Throughout next chapters, this work's methodologies will evidence not only that some physics visions can be applied to language but also that such application can be advantageous over purely statistical approaches of analysis and modeling that sometimes only scratch the surface of the problem.

Several parallels can be drawn between a linguistic system and a typical physical system. In Chapter 2 we elaborate on this similarities and we show how word embedding methods can take a key role in representing words in a flat space. The idea that in these systems it is possible to define an attractiveness metric emerges as the main motivation for this work. Furthermore, the metric of attractiveness is defined in a space where the complex structure of language is taken into account, each means, that the the global behaviour of the metric itself can be a relevant indicator to deal with natural language emergent properties.

1.3 Zipf's Law

Zipf's law is one of the most well known statistical regularities of linguistics. The law describes the empirical evidence that the frequency of a word decays as a power law of its rank, and so the rth most frequent word has a frequency $f(r)$ that scales according to

$$f(r) \propto \frac{1}{r^{\alpha}} \tag{1.1}$$

Equivalently, the law can be presented as a function of the frequency f of a word and becomes

$$P(f) \propto \frac{1}{f^{\beta}} \tag{1.2}$$

where $P(f)$ is the probability of a word having frequency f in a sample. In this work we will refer to the first form as the ranks distribution and to the second as the lexical spectrum. Although they may vary depending on the text, the exponents are usually $\alpha \approx 1$ and $\beta \approx 2$ [22].

Numerous derivations of Zip's law spread across different areas of science have been proposed using many formal ideas, frameworks, and sets of assumptions (see Piantadosi [23] for a critical review on Zipf's law proposed derivations). It is not within the scope of this thesis to address

or propose new explanations for Zipf's law, nevertheless evaluating text's Zipfian behavior is an important backing up procedure as one shall see in Chapter 2.

A note to say that some works claim that Zipf's law is perhaps an oversimplified empirical law and thus suggest other models that better fit the data like using a Yule-Simon distribution or considering two exponential regimes instead of one [24, 25].

1.4 Thesis outline

This thesis is divided in three main chapters of contributions. In Chapter 2, we go through the steps that led to the use of a word embedding model on a new role of computing a mass for each word distributed in a semantic vector space. In Chapter 3, we use an approach based on the definition of mass as a criterion for building a network of embedded words. In Chapter 4, we look into the possibility of establishing a quantitative measure of the informative content of a text. Such measure reflects a more intuitive notion of the term "information" when compared with Shannon's and Kolmogorov's frameworks.

In the three chapters a theoretical framework is provided followed by the methodologies used and finally the results with comparisons or parallels with previous works we found relevant.

The thesis ends with a Discussion and Conclusions chapter to summarize the work's take-home ideas, discuss the results and mention possible application or further work related with the ideas of this thesis.

Chapter 2

A Universe of Words

This chapter starts with the description, implementation and interpretation of word embedding word2vec model. Next, the results associated with a new word-property - the mass - show that it has advantages for use in macroscopic analysis of text.

2.1 Word Embedding

In recent years, the processing of natural languages had a growth due to the successful application of neural-based models that capture semantic properties of words, known as word embedding models. This models, based on distributed representation, provides simultaneously a way of quantitative representing word meanings as well as solving the *curse of dimensionality*.

The *curse of dimensionality* is a difficulty associated with modeling the joint distribution between many discrete random variables (such as words in a sentence). For example, to model the joint distribution of 10 consecutive words in a natural language with a vocabulary V of size $V = 100000$, there are potentially $100000^{10} - 1 = 10^{50} - 1$ free parameters which makes the direct derivation of a probability distribution from training data impractical. Besides, language models based on counting word sequences (N-grams) have massive limitations in generalizing for sequences not seen in training [26].

2.1.1 Distributional Semantics

The distributional hypothesis proposes to infer word meanings by watching the contexts they appear. As such, words that frequently appear in similar contexts are said to have similar meaning. This definition of meaning is therefore relative to other words and lies upon the creation of a similarity metric, $sim(w_1, w_2)$, between words that captures relationships such like

$$sim(\text{“}Monday\text{”},\text{“}Tuesday\text{”}) > sim(\text{“}Monday\text{”},\text{“}January\text{”})$$

Word embedding models achieve this by representing each word in the vocabulary as a feature vector in a real coordinate space of dimension N, $\mathbb{R}^N$ (where $N << V$).

In 2013, Mikolov et al. presented new neural-based architectures for unsupervised[a] learning that achieved state-of-the-art results in word embedding called *word2vec* [27, 28]. In this work we adapt Mikolov's architectures for our purposes of language macroscopic analysis.

2.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a powerful subset of machine learning algorithms. ANNs are based on a collection of connected units called the neurons which are generally grouped in layers (Fig. 2.1). A neuron receives a signal, processes it, and then passes the processed signal to the neurons connected to it. The connections have a weight, w_{ij} that is adjusted as learning proceeds and which increases or decreases the strength of the signal transmitted by each connection. The signal at a connection is a real value, x_i, and inside the neuron, occurs the sum all the inputs, having in mind their relative weight:

$$y_j^{'} = \sum_{i=1}^{m} w_{ij} x_i \tag{2.1}$$

[a] Word2vec is sometimes referred to as *self-supervised learning* method.

This weighted sum is then passed through a activation function to produce the output [29].

$$y_j = f(y_j^{'}) \tag{2.2}$$

Normally neurons of one layer connect only to neurons of the immediately preceding and immediately following layers. The data flows from the input layer and it is processed to generate a result in the output layer. In between them are zero or more hidden layers.

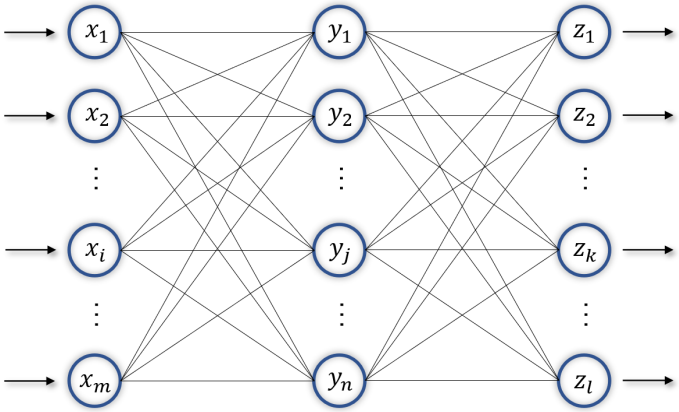

Figure 2.1: Diagram of an artificial neural network with fully connected layers.

Because of their versatility and ability to model nonlinear processes, artificial neural networks have found applications in many disciplines and in both supervised and unsupervised machine learning tasks [30, 31, 32].

2.2 Word2vec

Any Word2vec model relies on a three layer ANN that has a V size input layer fully connected to a N size hidden layer, which also fully connects to an output layer of size V (remembering that V is the size of sample's vocabulary while N is the arbitrary size of the embedding vector space). This kind of configuration where there are no intralayer connections between hidden and visible units is called a Restricted Boltzmann Machine (RBM) and it is an efficient method for learning a probability distribution over a set of inputs [33].

The input vector, $\mathbf{x}$, is a one-hot encoded vector, which means for a given input word, only one out of V units will be 1, and all other units are 0. The connections are done through a $\mathbf{M}_1$ weight matrix of size $V \times N$ (Input $\longrightarrow$ Hidden Layer) and a $\mathbf{M}_2$ weight matrix of size $N \times V$ (Hidden $\longrightarrow$ Output Layer) as portrayed in Fig. 2.2. The activation function of the hidden layer neurons is simply linear (i.e., directly passes its weighted sum of inputs to the next layer).

Each row of $\mathbf{M}_1$ represents a N-dimensional vector representation of the words in the vocabulary and so given an input word j, the hidden layer vector, $\mathbf{h}$, is

$$\mathbf{h} = \mathbf{x}\mathbf{M}_1 = \mathbf{w}_j^{M_1} \tag{2.3}$$

where $\mathbf{w}_j^{M_1}$ is the vector representation of word j in the $\mathbb{R}^N$ vector space of $\mathbf{M_1}$.

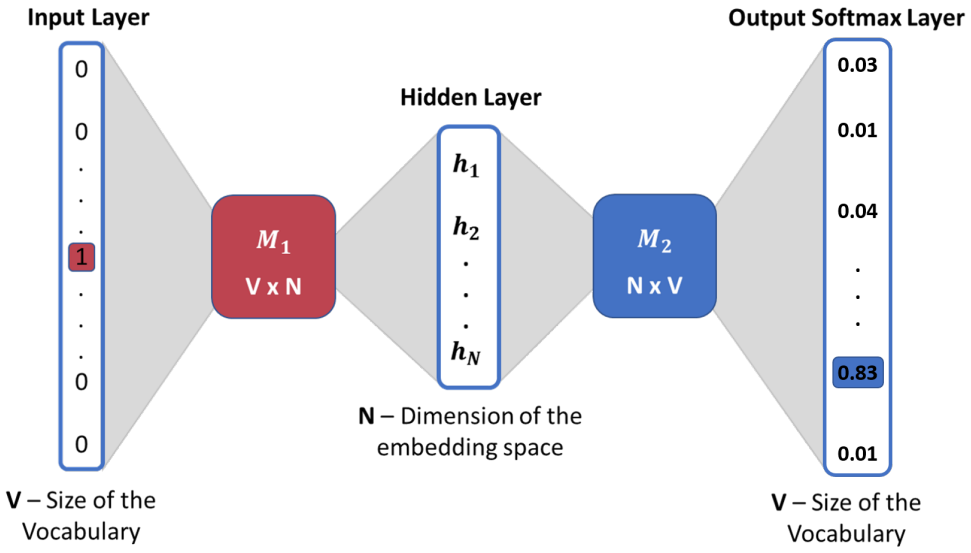

Figure 2.2: Diagram of word2vec's artificial neural network.

On the second weight matrix $\mathbf{M_2}$, each column corresponds to a second N-dimensional representation of the words $\mathbf{w}_k^{M_2}$.

The result of the product $\mathbf{w}_j^{M_1}\mathbf{M_2}$ is a V-dimensional vector. The output layer neurons then have a different activation function called a softmax function. This is so that the output value are interpreted as probabilities. In each of the k output neurons, the softmax function is[b]:

$$p(w_k|w_j) = \frac{exp(\mathbf{w}_k^{M_2} \cdot \mathbf{w}_j^{M_1})}{\sum_{k'=1}^{V} exp(\mathbf{w}_{k'}^{M_2} \cdot \mathbf{w}_j^{M_1})} \tag{2.4}$$

As one can conclude from the $p(w_k|w_j)$ notation, the V-dimensional output vector is interpreted as an array of probabilities of having each of the k output words conditioned to input w_j.

2.2.1 Training and Architectures

The training of the weight matrices is done by going through samples of text word-by-word and using surrounding words to predict a center word (*Continuous Bag-of-Word architecture*) or alternatively use a center word to predict possible surrounding words (*Skip-Gram architecture*) as pictured in Fig. 2.3.

More formally, let $S = \{w_1, w_2, w_3, ..., w_T\}$ be a sequence of words in a training sample with T words and c the size of the training context. To achieve the best performance in word prediction, the goal of each architecture is to maximize the average log probability

$$L_{CBOW} = \frac{1}{T}\sum_{t=1}^{T} \sum_{-c \leq j \leq c, c \neq 0} \log p(w_t|w_{t+j}) \tag{2.5}$$

[b] In equation 2.4, j and k are two generic indexes between 1 and V that respectively differentiate input and output words.

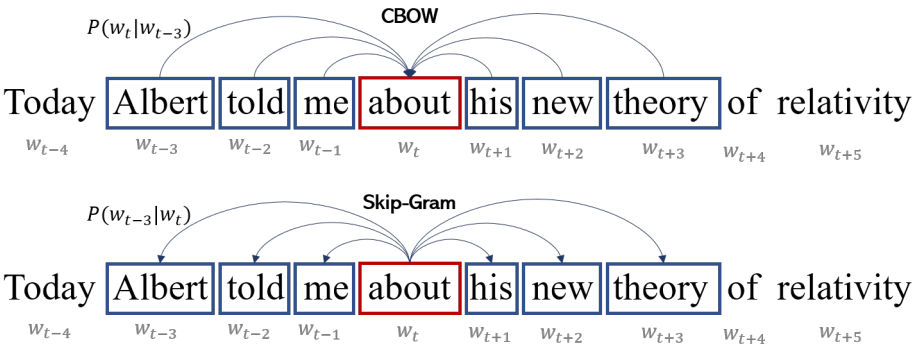

Figure 2.3: Portrait of *Skip-Gram* and *Continuous Bag-of-Word* (CBOW) architectures for a training context of size $c = 3$. Blue words form the context of the target word in red.

for the *Continuous Bag-of-Word* (CBOW) architecture, and

$$L_{SG} = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, c\neq 0} \log p(w_{t+j}|w_t) \tag{2.6}$$

for the *Skip-Gram* (SG) architecture. From equations 2.5 and 2.6 one gets the loss functions, E, for each architecture.

$$E = -L \tag{2.7}$$

Backpropagation is then applied to compute the gradient of E and consequently derive update equations for the weight matrices $\mathbf{M_1}$ and $\mathbf{M_2}$ using the stochastic gradient descent method. When efficiently trained, the hidden units model dependencies between the components of observations [34].

The derivation of the update equations was not included in Mikolov et al.'s original word2vec publications. Rong [35] published a more formal and detailed explanation of word2vec training process using the stochastic gradient descending method (see also [36] and [37] for further information on the topic).

2.2.2 Computational Optimizations

Both Skip-Gram and CBOW models can be very computational expensive since the cost of computing $p(w_k|w_j)$ is proportional to V which is often as large as 10^5 to 10^7 terms. Two approximation where proposed by Mikolov et al. [28] to tackle this problem: Hierarchical Softmax and Negative Sampling. In our work we opted for the more popular of the two which is negative sampling.

Negative Sampling

The idea for negative sampling comes from Mnih et al. [38] Noise Contrastive Estimation (NCE) which posits that a good model should be able to differentiate data from noise by means of logistic regression. In this context, the noise is a set of s words sampled from a noise distribution[c] $P_n(w)$. Negative sampling deals with the difficulty of having too many output vectors that need to be

[c]The noise distribution $P_n(w)$ is determined empirically. As described in [28], word2vec uses a unigram distribution raised to the $\frac{3}{4}$th power for best results.

updated per iteration by updating just $s+1$ vector per iteration (s negative samples plus one positive sample, i.e., the output word). With negative sampling, the training objective becomes the minimization of the loss function

$$E = -\log\sigma(\mathbf{w}_k^{M_2} \cdot \mathbf{h}) - \sum_{w_i \in \mathcal{W}_{neg}} \log\sigma(-\mathbf{w}_i^{M_2} \cdot \mathbf{h}) \tag{2.8}$$

where σ stands for the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and $\mathcal{W}_{neg} = \{w_1, ..., w_s\}$ is the set of words that are sampled based on $P_n(w)$, i.e., negative samples. Shortly, the training objective becomes to maximize the likelihood of the actual output word, $\mathbf{w}_k^{M_2}$, (first term in the right-hand side of 2.8) while simultaneously minimizing the likelihood for the i negative samples (second term in the right-hand side of 2.8).

This improvement made it feasible to train word2vec models in much larger sets and reach better embedding vectors. More complete descriptions of negative sampling, as well as the update equations, can be found in [36], [35] and [38].

2.3 Interpreting Word2vec

We mentioned before that each word in the vocabulary has it's N-dimensional representation in the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$. And since both input and output layers are V-dimensional, we can interpret $\mathbf{M}_1$ and $\mathbf{M}_2$ as two space transformation.

Let Ω and Γ be two vector spaces:

- Ω is the vector space in $\mathbb{R}^V$ where the words are represented as unique one-hot vectors. Such vectors do not capture any kind of semantic relationship between words. We call this the *text space*.
- Γ is the vector space in $\mathbb{R}^N$ where words are represented as reduced dimension vectors (embeddings). A metric of similarity $sim(w_1, w_2)$ represents semantic relations. We call this the *embedding space*.

The goal of word2vec neural network training is to iteratively adjust $\mathbf{M}_1$ and $\mathbf{M}_2$ so that

$$\mathbf{M}_1 : \Omega \to \Gamma \qquad\qquad \mathbf{M}_2 : \Gamma \to \Omega \tag{2.9}$$

Since the embedding space Γ is euclidean, the similarity between two vectors is given by the cossine similarity

$$sim(w_1, w_2) = \frac{\mathbf{w_1} \cdot \mathbf{w_2}}{\|\mathbf{w_1}\|\|\mathbf{w_2}\|} \tag{2.10}$$

Where $\mathbf{w_1}$ and $\mathbf{w_2}$ are the embedded vectors of words w_1 and w_2. In words2vec, the embedded vectors are more often the rows of $\mathbf{M}_1$, however, some users report better results in particular tasks when using the columns of $\mathbf{M}_2$ for the representations or a combination of both $(\mathbf{M}_1 + \mathbf{M}_2^\intercal)/2$. Both matrices capture semantic relations [39].

The value of $sim(w_1, w_2)$ is bounded in $[0, 1]$ so a well trained embedding model should produce

relations like $sim(\text{"}Monday\text{"},\text{"}Tuesday\text{"}) \approx 1$ or $sim(\text{"}Monday\text{"},\text{"}Goalkeeper\text{"}) \approx 0$.

2.4 Words' Mass

Traditional statistic that takes place in text is carried out in *text space* Ω where all words are indistinguishable as interactions between them are not taken into account. Zipf's law, for example, is one of language's statistical regularities observed in Ω which is seen as evidence of emergent behavior of natural languages. But knowing that the Zipf's law comes only from the frequency of words in a sample, it remains totally valid even if the sample is randomly shuffled. Therefore it is not strictly related with text's structure.

The idea behind implementing word2vec is to perform macroscopic analysis in Γ instead of Ω. But while in Ω statistics is done over individual word frequencies or N-gram sequences, in Γ such information is lost. The *embedding space* Γ is totally defined by the information in matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ from word2vec.

One of the main motivations of this work is to define a metric for words in Γ that reflects word's "*attractiveness to use*" in a similar way frequency does in Ω but without both being interdependent. The key is that this new quantity, which we call *mass* [d], is based on semantic relations between words so that one can perform macroscopic analysis on text without bypassing text's complex structure.

2.4.1 Defining Mass

Knowing that the mass of each word must be defined simply using $\mathbf{M}_1$ and $\mathbf{M}_2$, the most intuitive approach was to use the assumptions of equation 2.2 which is recalled here

$$p(w_k|w_j) = \frac{exp(\mathbf{w}_k^{M_2} \cdot \mathbf{w}_j^{M_1})}{\sum_{k'=1}^{V} exp(\mathbf{w}_{k'}^{M_2} \cdot \mathbf{w}_j^{M_1})} \tag{2.11}$$

The denominator on the right side of the equation is simply a normalization factor so that the result can be interpreted as a probability. It is the exponential in the numerator that quantifies the likelihood of the words w_j and w_k being found together. Remembering, from eqs. 2.6 and 2.5, that word2vec maximizes $p(w_k|w_j)$ for every training pair, which in practice means maximizing $exp(\mathbf{w}_k^{M_2} \cdot \mathbf{w}_j^{M_1})$ or, even more simply, maximizing $\mathbf{w}_k^{M_2} \cdot \mathbf{w}_j^{M_1}$. Hence two words w_j and w_k that frequently appear together in text ("hong" and "kong" for example) have high dot product $\mathbf{w}_k^{M_2} \cdot \mathbf{w}_j^{M_1}$.

On this basis we define the quantity *mass* in Γ as

$$\boxed{M_{w_i} = \sum_{j=0}^{V} m_{ij} \qquad \text{with} \qquad m_{ij} = \begin{cases} 1\,, & \mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} > 0 \\ 0\,, & \mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} \leq 0 \end{cases}} \tag{2.12}$$

[d] The choice of the word "mass" is justified because it is a measure, even if abstract, of attractiveness.

Where the scalar M_{w_i} stands for the *mass* of word w_i, not be confused with matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ from word2vec.

This definition simply implies evaluating the dot products of w_i represented in $\mathbf{M}_2$ with all the V words in the vocabulary represented in $\mathbf{M}_1$ and *counting* how many are greater than 0. The reason for choosing the threshold 0 for m_{ij} is because before training, every $\mathbf{w}_j^{M_1}$ and $\mathbf{w}_k^{M_2}$ vector is initialized as a random unit vector in $\mathbb{R}^N$ and the mean value of the inner product between two unit vectors is 0 (see appendix A). As training proceeds, words w_j and w_k that frequently appear together will result in $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} > 0$ as the opposite, $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} < 0$ will result for words that don't frequently share context.

In the following sections, we will explain our methodology for evaluating mass that is naturally preceded by an implementation of a word2vec model, and we will compare the results of statistical analysis over masses M in *embedding space* Γ with the same analysis over frequencies f in *text space* Ω.

2.5 Methodology

The entire implementation was done in Python 3. In this section we present the steps from gathering the corpus, passing through word2vec, and ending in the evaluation of the masses defined by 2.12.

2.5.1 The Corpus

Like any machine learning algorithm, word2vec requires a training set, which, in such case, is a large amount of text - a *corpus*. The size of the corpus used is somewhat arbitrary but larger corpus means more training examples and therefore better generalization. Corpus used to train word2vec are often as large as 10^6 to 10^{11} words. Research indicates that on more subject-specific corpus, good embeddings can be achieved with smaller sets [40].

For this work's corpus we use a publicly available compilation of news articles and economic reports from the 2015-2020 period gathered by Bernardo [41]. The corpus contains about 14 million words. Hereafter it is referred to by the abbreviation 14MWC (*14 million word corpus*).

2.5.2 The Model

To implement word2vec we used the Gensim Python library which, as far as we know, has the most memory efficient open-source implementation of the words2vec algorithm.

Corpus Pre-Processing

To train the model, the corpus must be pre-processed and arranged as a list of sentences, where each sentence is a list of words. Our processing also included removing non-alphabetical or numerical characters and shift capital characters to lowercase. The pre-processing is shown in the following code snippet.

```
from nltk import sent_tokenize
from nltk.tokenize import word_tokenize
import re

# Open File
file = open('corpus.txt', encoding='utf-8', errors='ignore')
book = file.read()
file.close()

raw_sentences = sent_tokenize(book)

def word_tokenizer(raw):
    #removes non-alphabetic or numerical characters
    clean = re.sub("[^a-zA-Z]"," ", raw)
    words = word_tokenize(clean.lower()) #makes all characters lowercase
    return words

sentences = []
#processes text to a list of lists
for raw_sentence in raw_sentences:
    if len(raw_sentence) > 0:
        sentences.append(word_tokenizer(raw_sentence))
```

Model Implementation

The model initialization and training were combined in a function `run(sentences, cycles, dim, architecture, context)` where variable `sentences` is the list of sentences from the pre-processing, `cycles` is the number of iterations over the corpus, `dim` is the dimensionality of the embedding space, `architecture` selects the model architecture (1 for Skip-Gram; otherwise CBOW) and `context` is the size of the context window.

Although relatively arbitrary, a value around $N = 300$ is recommended for dimensionality in word2vec's original publications as choosing higher values does not necessarily improves the model [27, 28]. However, it is a hyperparameter that is advised to be tuned to the corpus in use [42]. The same goes for the `cycles` hyperparameter as smaller corpus tend to benefit from more iterations. The following code shows the model building.

```
import gensim.models.word2vec as w2v

def run(sentences,cycles,dim,architecture,context):
""" sentences - list of lists of training sentences
    cycles - number of cycles trough "sentences"
    dim - dimension of the embeddings
    architecture=1 for Skip-Gram, architecture=0 for CBOW"""

    Model = w2v.Word2Vec( #initialize model
        sg = architecture,
        workers = multiprocessing.cpu_count(),
        size = dim,
        window = context, #context size
```

```
14          hs=0, #implements negative sampling optimization
15          sample = 0, #no subsampling
16      )
17
18      Model.build_vocab(sentences) #build vocabulary
19
20      #train the model
21      Model.train(sentences,epochs=cycles,total_examples=Model.corpus_count,
22      start_alpha=0.01, end_alpha=0.0005)
23
24      return Model
25
26  Model = run(sentences,10,300,0,5) #Model creation
```

Users familiar with word2vec may know another optimization referred by Mikolov et al. [28] as *Subsampling of Frequent Words* that sugests subsampling words that have very high frequency (e.g. "in", "the", "a") for speed purposes. Subsampling would alter the words frequency in text and, in our specific task, that would be reflected on the mass. Therefore no subsampling is applied as the parameter `sample=0` (line 15) indicates.

Full documentation for Gensim's word2vec implementation and hyperparameters can be found in [43].

Measuring Mass and Frequency

The function `run` returns an object `Model` that contains the trained matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ and the list of words in the vocabulary, `Model.wv.index2word`.

The following code snippet shows the implementation of equation 2.12 for evaluating the masses (line 13) well as the frequencies (line 14).

```
1   import numpy as np
2
3   top_n_words = 40000
4
5   M1 = new_Model.wv.syn0[0:top_n_words]
6   M2 = new_Model.syn1neg[0:top_n_words]
7
8   frequencies=[]
9   masses=[]
10
11  for i in range(top_n_words):
12
13      masses.append( sum( np.dot(M2[i], M1.T) > 0) )
14      frequencies.append(Model.wv.vocab[Model.wv.index2word[i]].count)
```

The variable `top_n_words` means that instead of evaluating the mass of all words in the vocabulary, we evaluated only the subset of the n most frequent words. Doing this has two advantages:

- The evaluation of V masses implies a number of operations that scales with V^2. The 14 million corpus has $V \approx 150000$ and evaluating all the masses takes between 1-2 days using

a standard quad-core laptop computer. By considering a subset of the $n = 40000$ most frequent words the time is reduced to a couple hours.

- Low frequency words are not well fitted by the model as they are present in few training examples. Consequently the vector representations of these words are much less significant since they are little updated from its initial random state. Not considering these words is a way to reduce the noise in the mass evaluation.

Model Validation

To ensure that our word2vec model was working correctly, we used two word embedding intrinsic evaluation methods. It was found that both for Skip-Gram and CBOW architecture the evaluation results are in line with other models trained in corpus of similar dimension. The evaluation methods and results are shown in detail in appendix C.

2.6 Results

2.6.1 Mass vs. Frequency

We first test our definition of mass by comparing it with word's frequency: As expected, our results show that mass and frequency are positively correlated as perceived in Fig. 2.4. The results were consistent in showing that the correlation is higher when using the CBOW architecture. With CBOW, the Pearson's correlations between mass, M, and frequency, f, where found between $r_{M,f} = 0.38$ and $r_{M,f} = 0.53$ depending on the set of hyperparameters while with Skip-Gram the numbers drop to between $r_{M,F} = 0,07$ and $r_{M,F} = 0,20$.

Mass is somehow a measure of the number of context where the word appears consistently. So one can expect that words with high frequency will also appear in more contexts not just because of statistics, but as they are also more generic meaning words (*the*, *and*, *of*, *a*, ...) and therefore have higher mass. While on the contrary, words with lower frequencies tend to appear in fewer and more specific contexts. The hypothesis that the frequency inversely relates to words' semantic content is well known. A compelling test of this statement is high frequency words are the first to be suppressed in telegraphic speech [44]. Our correlation results indicate that the same hypothesis must be valid for the masses.

The consistent observation of positive correlation between mass and frequency is itself a noteworthy result as it shows that a concept of *attractiveness to use* (frequency in the *text space* Ω) is passed to *embedding space* Γ by the transformations $\mathbf{M}_1$ and $\mathbf{M}_2$. Thus, the vector space Γ, in addition to being provided with the important similarity metric between words, is also provided with metric of attractiveness - *the mass*.

Despite the fact that we refer to it as a measure of the number of context in which a word appears, it is important to point out that it is a quantity that must be taken as relative and not absolute. That is, it is not accurate to say that a word appears in a number of contexts equal to its mass (especially because the definition of context can be itself vague). To our understanding, mass is essentially a relative metric of attractiveness to use or semantic specificity (both inversely

related).

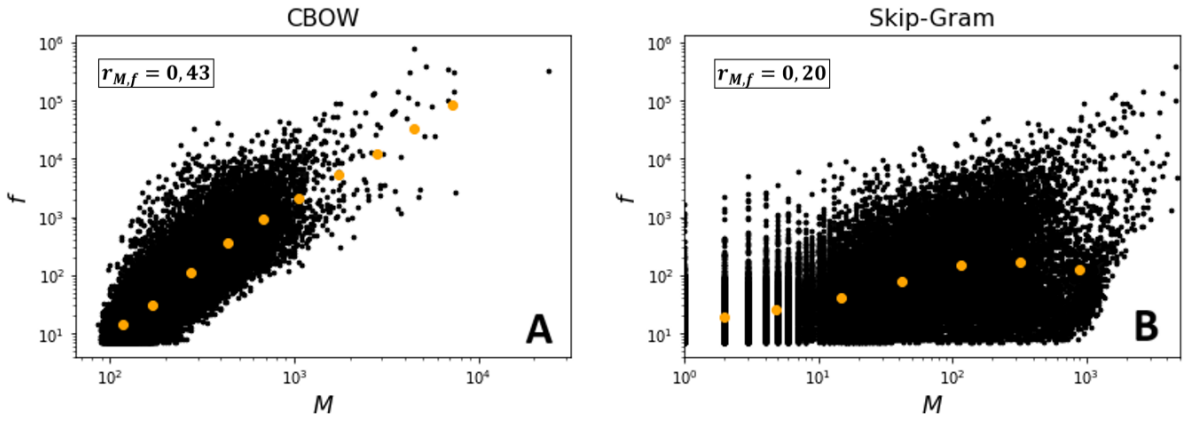

Figure 2.4: Mass and frequency relation for the 40000 most frequent words in the corpus. $r_{M,f}$ is the Pearson's correlation between frequency f and mass M (black points). (A) word2vec with CBOW architecture; (B) word2vec with Skip-Gram architecture. Both models built with dimensionality $N = 200$ and window size $c = 5$. Orange points are the average of log binned data and indicate tendencies. Tests with different sets of hyperparameters were consistent in revealing that the CBOW architecture results in higher $r_{M,f}$ comparatively to Skip-Gram.

It is not clear to us why the Skip-Gram and CBOW architectures produced significantly different correlation results. In theory the definition of mass would be equally compatible with any of the architectures and the results of the model evaluation even indicated that skip-gram models score better in embedding evaluation tasks. However, not only the CBOW $r_{M,F}$ were consistently higher, as also some skip-gram models often produced very low $r_{M,F}$ or large amounts of words with $M = 0$. We believe that the observed differences are related to the optimizations for multi-word contexts that differ for each architecture [45]. Nevertheless, CBOW as generally proven to be the most suitable architecture to achieve word mass in line with the expectations.

2.6.2 Mass' Zipfian Behaviour

Once established that mass and frequency are related, we also compare how they are distributed. This is particularly relevant knowing that word frequencies follow a power law known as Zipf's Law which we introduced in section 1.3.

In Fig. 2.5 we have the frequency distribution of the words in our corpus. Both rank distribution and the lexical spectrum (histogram) follow an approximated power-law which implies that the curves are approximately linear on log-log scale.

Fig. 2.6 shows the rank distribution and lexical spectrum of the masses for both Skip-Gram and CBOW architectures. The results reveal a power-law decay for the mass distribution. Yet the power laws were consistently more evident when using the CBOW architecture and some of the trials with Skip-Gram simply did not exhibited a power-law. The CBOW trials, however, consistently lead to distribution that can be classified as Zipfian.

Note that in the lexical spectrum it was sometimes observed that the curve clearly presented two decay regimes where the tail of the high masses decays with a higher exponent as represented in the figures 2.6B and 2.6D. Previous work by Cancho et al. [24] approached frequency distribution as a two regime power-law instead of only one. The authors state that the two observed exponents divide words in two different sets: a kernel lexicon formed by a small number

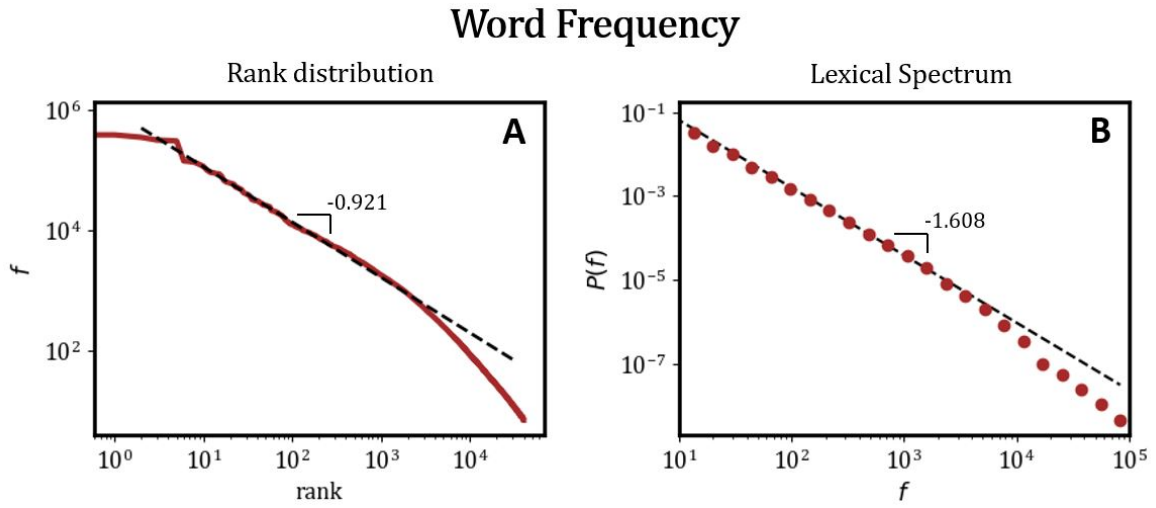

Figure 2.5: Word frequency distribution for the 14MW corpus. (A) rank distribution. (B) Frequency Histogram (lexical spectrum). Both curves fit power laws as predicted by Zipf's law. Dashed lines are power-law fits with the respective exponent.

of versatile words and an unlimited lexicon for specific communication. Although it is not in the scope of this thesis to draw linguistic-related conclusions, our results clearly support this hypothesis not only because the two regime decay is evident, but most important because the results hold on vector space Γ where semantic and syntactic relations are represented.

2.6.3 Mass does not exist on shuffled text

We subjected our definition of mass to one last simple but relevant test. The idea was to evaluate the mass of words in the same corpus with the words randomly shuffled. Note that shuffling implies keeping the frequencies unchanged but completely destroying the text structure and making it unreadable. The shuffling was done by the `total_randomize_sentences()` method described in the appendix D.

We verify that with the shuffled corpus, almost all words have mass zero. To be precise, when considering the subset of the 40000 most frequent words, more than 99.9% of the words in all trials register mass M=0 and the remaining register mass M=1 or M=2. If we use the total 14 million words instead of the just 40000 most frequent, then each word registers a mass around M=300 which is due to very infrequent words whose vector were practically not updated from it's initial random state. Therefore, we verify that using a subset of the most frequent words has effects on minimizing the noise in the mass results.

If mass does not exists on shuffled text it means that the quantity is only relevant on structured text and it does not bypasses it's complex emergent structure. Furthermore, the Zipf's law of frequencies holds in shuffled text, that is, it does not necessarily imply a real text that follows semantic and syntactic rules. The mass distribution, however, does not exist in shuffled text. With this we propose that evaluating the mass Zipfian behaviour Therefore, it is important to consider a subset of the most common words to minimize the noise in the mass results.

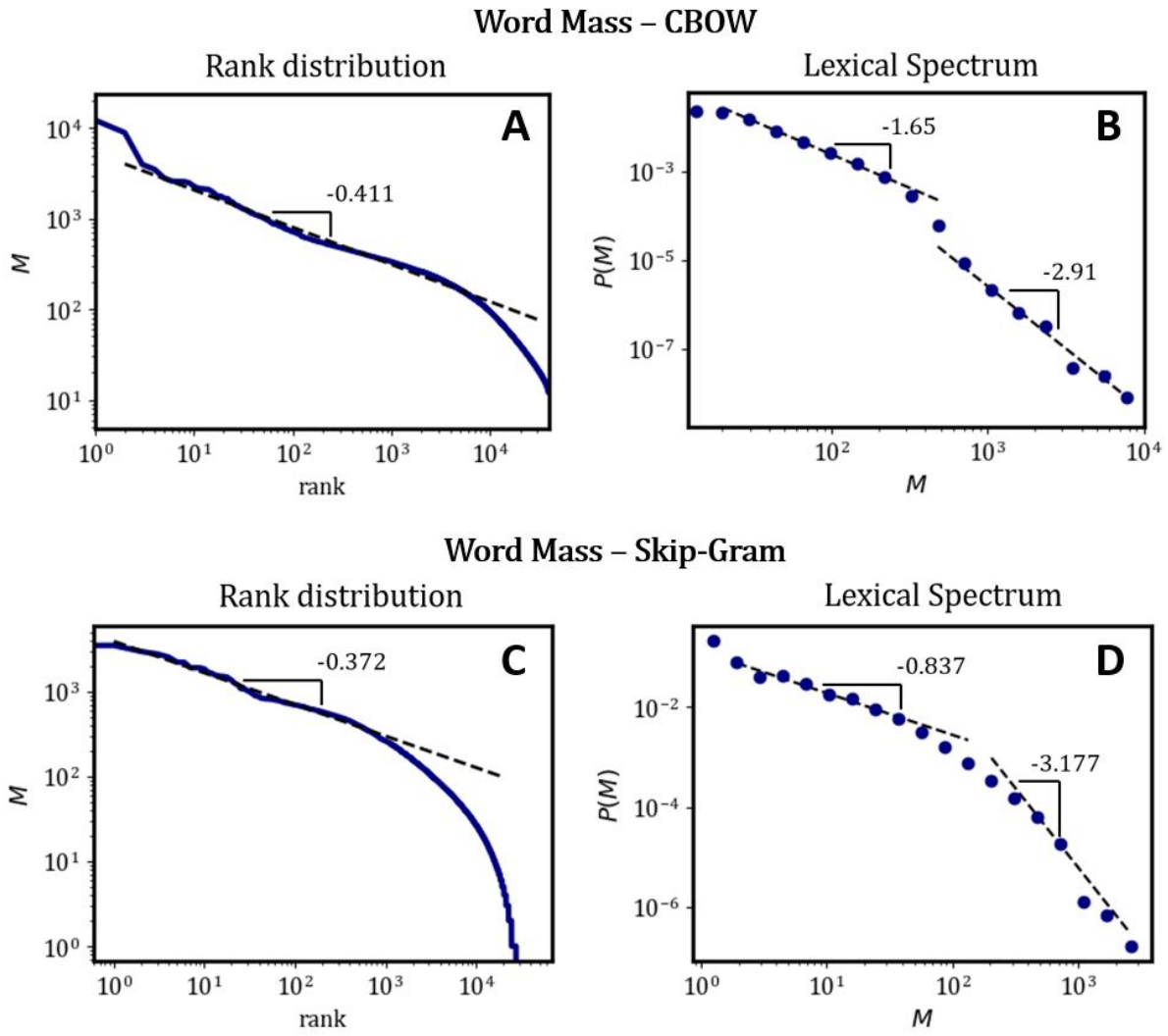

Figure 2.6: Mass distributions from CBOW and Skip-Gram model. Both models with dimensionality $N = 200$ and window size $c = 5$. (A) Rank distribution CBOW. (B) Frequency Histogram (lexical spectrum) CBOW. (C) Rank distribution Skip-Gram. (D) Frequency Histogram Skip-Gram Dashed lines are power-law fits with the respective exponent.

The result also supports the previous ones from sections 4.3 and 4.4 as it demonstrates that mass and frequency are somewhat independent quantities that only correlate when the text follows a certain organization. To explain it we can recall the transformations:

$$\mathbf{M}_1 : \Omega \to \Gamma \qquad\qquad \mathbf{M}_2 : \Gamma \to \Omega \tag{2.13}$$

The very conception of these transformations implies that the space Ω of dimension V can be transformed into a space Γ of dimension N (where $N << V$). For dimension reduction to be feasible without loss of information, the Ω space must necessarily have redundant dimensions. The number of dimension of Γ (between 100 and 500) is reduced precisely so that the space is constructed without having redundant dimensions; and construction is feasible since the structure of regular text is not random. When the text is shuffled the randomness increases considerably since local syntactic and semantic relationships are broken. The consequence of this randomness is that the Ω text space of size V is no longer redundant enough to be described in any

N-dimensional space without loss of information. Where in such case the information is encoded in the value of the mass. In Chapter 4, we elaborate on this subject to suggest that the mass can also serve as a measure of information.

Chapter 3

A Networks of Embedded Words

This chapter begins with a critical exposure to existing linguistic networks - networks models of the human languages as a complex system - and leads to the the introduction of a new form of unsupervised linguistic network.

3.1 Complex Networks

Complex networks are essentially graph-like structures of interconnected elements [46]. The elements are called the nodes (or vertices) and the connections, that represent interaction between elements, are called the edges (or links). In its simplest form, the edges of the network are undirected and unweighted (Fig. 3.1 A). They can also be directed if the direction of the edges is relevant to the interaction - like in a buyer-seller relationship of an economic system (Fig. 3.1 B). When the connections have different strengths, the network is a weighted network (Fig. 3.1 C).

Complex networks are used to describe a wide variety of systems from the World Wide Web [47], to protein structures [48] and languages. An appropriate use of network analysis depends on the right choice of network representation. The same elements can generate different networks depending on the connection criterion. In a network of individuals, for example, the edges can represent a variety of relations like professional collaborations [49], sexual contacts [50] or Facebook friendships [51].

With research in linguistic networks, new quantitative approaches for understanding human language as a system were developed. In addition, the framework of complex networks places linguistic research in broader and interdisciplinary context [52]. In section 3.2 we will discuss some previous works on linguistic networks to compare with a new linguistic network using the concepts introduced in chapter 2.

3.1.1 Characterizing a Network

A possible mathematical representation of a network is by using an adjacency matrix. A system of N nodes has an adjacency matrix of size $N \times N$. The elements of the matrix a_{ij} represent the edges and they take the values $a_{ij} = 1$ (if there is a connection between node i and node j) or $a_{ij} = 0$ (absence of connection) in an unweighted network. In a weighted network the elements a_{ij} take real value. Here, we can introduce three important macroscopic properties used to characterize each complex network: Degree distribution, average path length and clustering coefficient.

The degree k of a node is the number of edges connected with that node. For an unweighted and undirected network:

$$k_i = \sum_{j=0}^{N} a_{ij} \qquad , \quad a \in \{0,1\} \tag{3.1}$$

For directed network each node has an in-degree k_i^{in} and an out-degree k_i^{in} which are the number of edges that goes in and out of that node i respectively. For weighted networks the degree is extended directly to the strength (or weighted degree) s, which is the sum of the weights of all edges connected to the node [53]

$$s_i = \sum_{j=0}^{N} a_{ij} \qquad , \quad a \in \mathbb{R} \tag{3.2}$$

From the degrees of the individual nodes, one can find the probability $P(k)$ that a randomly

selected node has exactly degree k. It was found that, for a large number of real-world network, the degree distribution follows a power-law

$$P(k) \propto k^{-\lambda} \tag{3.3}$$

Such networks are called scale-free [54].

The average path length between two nodes is defined as the average of the shortest paths between any two nodes:

$$L = \frac{2}{N(N-1)} \sum_{i,j} d_{ij} \tag{3.4}$$

where d_{ij} is the shortest path length between i and j, defined as the minimum number of edges traversed to get from node i to node j. For weighted networks, the weighted average path length is the same as in 3.4 but with d_{ij} defined as the smallest sum of the weights of the links throughout all possible paths from node i to node j.

Finally, the clustering coefficient is a measure of the degree to which nodes in a network tend to cluster together. Defined by Watts and Strogatz [55], the clustering coefficient C_i of node i is

$$C_i = \frac{2E_i}{k_i(k_i-1)} \tag{3.5}$$

as $k_i(k_i-1)/2$ is the maximum number of edges between the first neighbours of node i and E_i is the actual number of edges existent between such nodes. The clustering coefficient, C, of the whole network is defined as the average of C_i.

Although there are more network-related quantities, $P(k)$, L and C are three robust measures of a network's topology [46]. A final reference to the small-world coefficient S^{Δ} which, by Humphries et al. [56] definition, is obtained from the quantities C and L.

$$S^{\Delta} = \frac{C/C_{rand}}{L/L_{rand}} \tag{3.6}$$

where $C_{rand} \approx \frac{\langle k \rangle}{V}$ and $L_{rand} = \frac{\ln V}{\ln \langle k \rangle}$ are, respectively, the clustering coefficient and average shortest path length of an Erdös-Rényi random network [57, 58] with same size, V, and average degree, $\langle k \rangle$.

The term small-world designates networks where the nodes are separated by a small number of edges from each other, when compared with the total size of the network. More precisely, for a small-world network the average shortest path L grows proportionally with the logarithm of the number of nodes N, that is:

$$L \propto \log N \tag{3.7}$$

Examples include airport networks [59], social networks [60] or brain neuron networks [61]. Small world networks are characterized by a low average path and a clustering coefficient significantly higher than one would expect from a random graph. Definition 3.6 is intended to reflect these characteristics and thus a network is considered to be small-world if $S^{\Delta} > 1$.

3.2 Linguistic Networks

Linguistic networks are networks models of the human language. In such networks, the nodes represent the words from a language vocabulary whereas the edges represent pairwise relations of a particular type between these words. The linguistic network approach is a way of modeling language that allows for a better understanding of some of the most debated issues related to language. Namely the way the brain stores and accesses words in an extremely efficient way [62, 63] or the description and origin of the emerging structure of natural languages [64]. There are also several practical applications of linguistic networks in natural language processing tasks [65, 66].

Linguistic Networks	
Co-occurrence Network	Two words are joined by an edge if they co-occur (i.e., they are adjacent) somewhere in a sample of text. [67]
Syntactic Dependency Network	Words are joined if they form a syntactic dependency relations in at least on sentence of a text sample. Syntactic dependency refers to a *dependent* word that in some way modifies a *head* word (eg. noun-adjective and verb-subject relations). The construction of such networks requires the corpus to be tagged manually. [68]
Lexical Networks	A broader category of linguistic networks where edges may be based on semantic, orthographic or phonological features of the words [66]. Lexical networks include Semantic similarity networks [63], Synonyms [69] network, Orthographic similarity networks [70] , Phonological similarity [62] networks and others [71, 72]. Such networks are not constructed based on a corpus like co-occurrence and syntactic dependency networks.

Table 3.1: Types of linguistic networks.

Multiple types of linguistic networks have been constructed to capture specific properties of the human language, being semantic or syntactic. A summary is shown in table 3.1. Lexical networks[a] capture properties of words (meaning, spelling, etc.) but their construction process is not based on a corpus. On the other hand, co-occurrence and syntactic dependency networks are built directly from samples of text and therefore are more suited to examine structural features.

Constructing a co-occurrence network is a totally unsupervised process as edges link words that co-occur (i.e., are adjacent) in a text sample. Fig. 3.1 illustrates co-occurrence networks based on tree sentences. In syntactic dependency networks the edges indicate dependency relations where some word is modified by another (noun-adjective and verb-subject relations for example). Two words are connected if they form a syntactic dependency relation in a sentence of the text sample. Unlike co-occurrence networks, the construction of syntactic dependency networks is supervised as the dependencies need to be annotated manually.

[a]Considering lexical networks as a category of linguistic networks that includes semantic, orthographic and phonological networks is not a nomenclature adopted by all authors. Here it is used the same way as in [66].

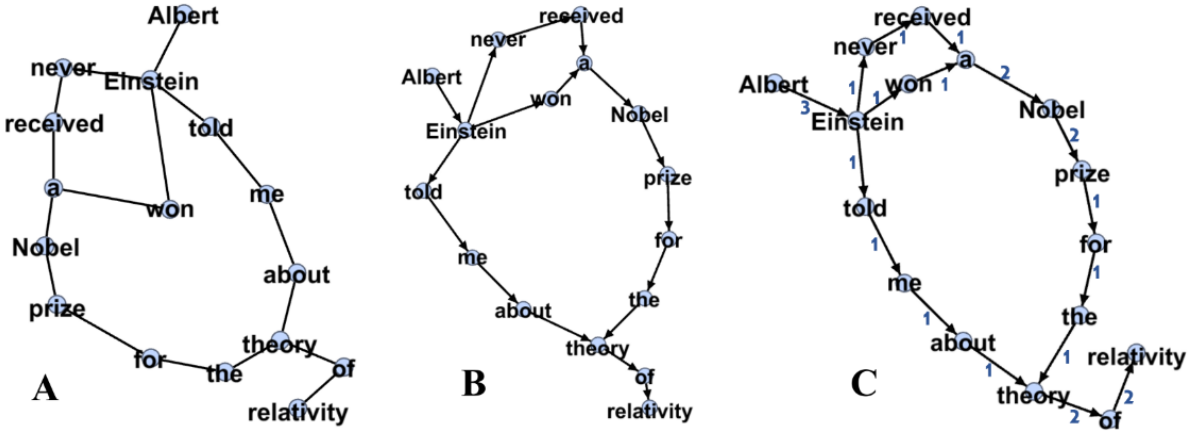

Figure 3.1: Word co-occurrence networks based on three sentences: (1) *"Albert Einstein told me about theory of relativity"*; (2) *"Albert Einstein won a Nobel prize."*; (3) *"Albert Einstein never won a Nobel prize for the theory for relativity."*. (A) Undirected and unweighted network; (B) Directed and unweighted network. The direction of the edges represent words' order in text; (C) Directed and weighted network. The weight of the edge is simply the number of times the relation repeats in the sample of text.

As previously mentioned, this work follows Saussure's early ideas that language is a system in which each linguistic unit is defined by, and only by, its relations with the other units. And even though finding those relations is not trivial, the idea implies that syntactic and semantic information can be retrieved in an unsupervised way, i.e. without tagging, annotations or other information apart from the sequence of words that form the text. In chapter 2 we saw with word2vec that a sense of meaning can be effectively found with unsupervised learning.

Of the set of proposals to represent text's structure in the form of a network, the only one that proposes to do it in an unsupervised way is the co-occurrence network.

3.2.1 The Problem of the Co-occurrence Network

The word co-occurrence network was first introduced by Ferrer i Cancho et al. [67] as an undirected and unweighted network that links adjacent words in text. Following works proposed adding direction and weight to the edges so the the network could reflect more information about the system: The direction would reflect the ordering of the words in a co-occurrence pair and the weights would reflect the frequency with which the pair was found.[73]

It was found that whatever the type of edges and text sample used (provided it was large enough), the degree distribution $P(k)$ of the co-occurrence network follows a power law making it a scale-free network [67, 73, 74]. A result that, due to the power-law decay and it's apparent universal character, has evident touching points with Zipf's law for word frequencies (eq. 1.2). Even more, noting that sometimes the degree distribution, as well as Zipf's law, appears with two power law decay regimes. An effect that, according to Choudhury et al. [75], becomes more evident in larger corpus. As expected, it was also verified that the frequency of words in the text and the degree of the corresponding node in the network are two highly correlated measures, since a word that repeats more often also tends to have more different adjacent words [67].

Following the growing interest in linguistic networks, Masucci et al. [74] investigated the topological differences on co-occurrence networks constructed with normal text and shuffled text (where

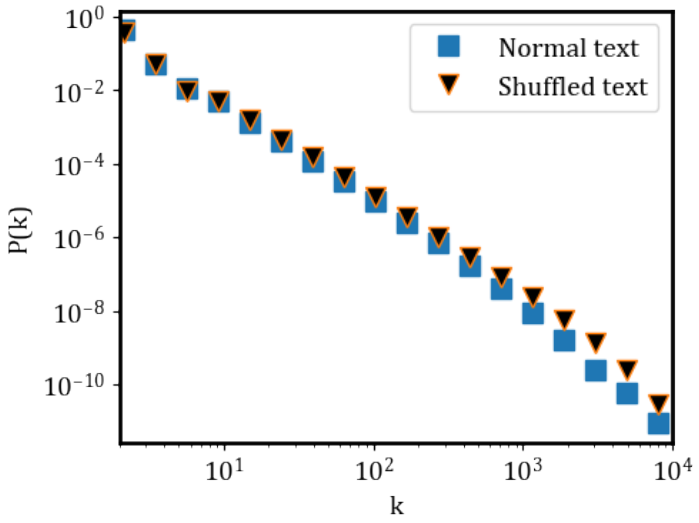

Figure 3.2: Degree distribution of word co-occurrence networks build with this work's 14MWC. The degree distribution is practically unchanged by the shuffling of the corpus.

the shuffling process preserves word frequencies). The results unmistakably showed that the macroscopic features of the network - degree distribution (illustrated in Fig. 3.2), the strength distributions, average cluster coefficient and others - remain virtually unchanged when the text is shuffled.

The results make it clear that the degree distribution is a consequence of the frequencies and unrelated to the text structure: Keep the frequencies of words and, on the macroscopic plane, the network remains practically unchanged whether the text is readable or not.

In the same work, Masucci et al. proposed a new measure to distinguish the real text network from the shuffled network. The measure was named *vertex selectivity*, e, and it "captures the effective distribution of numbers in the weighted adjacency matrix"[74]. The selectivity of node i is defined as

$$e_i^{in/out} = \frac{s_i^{in/out}}{k_i^{in/out}} \tag{3.8}$$

The plots in Fig. 3.3 show that selectivity distribution is altered by the shuffling of the text. However, two limitations are identified regarding what vertex selectivity can tell about the structure of the text: (1) The measure can only be applied to weighted nets, as it depends on the node strength, s. (2) As shown in Fig. 3.3, the distribution of the selectivities is not fully altered since for both normal and shuffled text it appears to follow a power law. Only the characteristic exponent is changed.

It is easy to see that it can be problematic to study the structure of the text through the co-occurrence network, especially in unweighted networks where it is not possible to take into account the selectivity. If the network is almost entirely defined by the frequencies of the words then the discussion of the network topology turns out to be the same as the one around the origin of the Zipf's law.

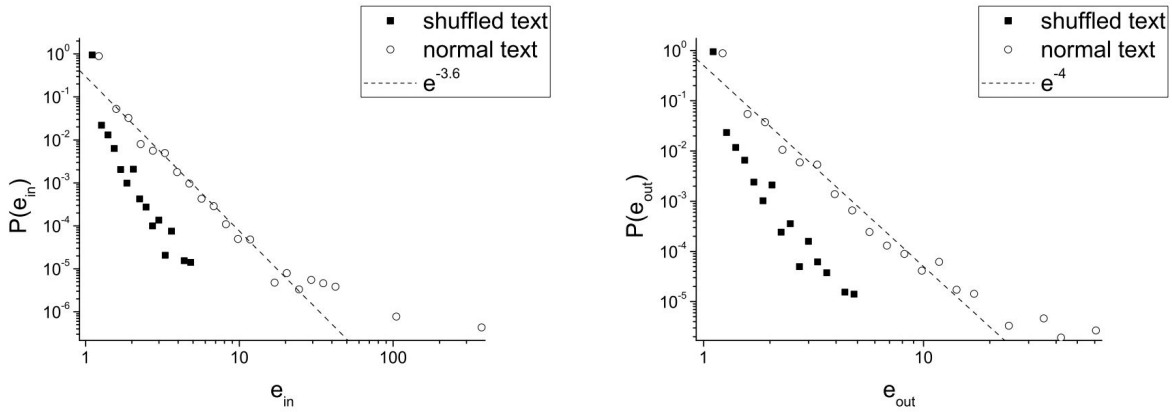

Figure 3.3: The out and in-selectivity distributions of *Moby Dick* compared to the distributions obtained after shuffling the text. From [74] with permission from the author.

3.3 A New Embedding-Based Network

Following the ideas proposed in Chapter 2, we developed a new type of unsupervised linguistic network based on the definition of mass, M, which is recalled here:

$$M_{w_i} = \sum_{j=0}^{V} m_{ij} \qquad \text{with} \qquad m_{ij} = \begin{cases} 1\,, & \mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} > 0 \\ 0\,, & \mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} \leq 0 \end{cases} \tag{3.9}$$

The idea for our network is that the edges reflect the condition m_{ij} from eq. 3.9. Based on this criterion, we propose that the network can be constructed in the following ways:

Undirected and Unweighted: Node i connects to node j if $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} > 0$ or $\mathbf{w}_i^{M_1} \cdot \mathbf{w}_j^{M_2} > 0$.

Directed: Node i connects to node j with direction $i \to j$ if $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} > 0$.

Directed and Weighted: Node i connects to node j with direction $i \to j$ if $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2} > 0$. The weight of the edge $i \to j$ is $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2}$

Where one must recall that the notation $\mathbf{w}_i^{M_1}$ stands for the vector representation of word i in matrix $\mathbf{M_1}$ of word2vec's neural network.

We call this the *embedding network* as its construction requires the text to pass by the word2vec's word embedding algorithm. Also note that word2vec is an unsupervised method and therefore no tagging, annotation or *a priori* knowledge of the text is required to build the network.

In case the network is directed, the out-degree of node i (the number of outgoing edges) is exactly equal to the mass of that word in the embedding space Γ. An important note regarding the dimensionality of the representations: In chapter 2 we saw that word2vec operates on the *text space* Ω of size $V \times V$, effecting a dimensionality reduction to the *embedding space* Γ of size $V \times N$. As we have seen, a complex network is fully described by its adjacency matrix $V \times V$

however this does not mean that this is the real dimensionality of the system since the adjacency matrix of the embedding network is totally built with the information from matrices $\mathbf{M_1}$ and $\mathbf{M_2}$ of word2vec which are of size $V \times N$.

In terms of represented features, the proposed embeddings network and the co-occurrence network have points in common:

- Words with higher frequency in text tend to have higher degree in the network as we showed in section 2.6 that frequency and mass - and therefore the degree - are positively correlated.
- The edges link words that appear together in text. As mentioned in section 2.4.1, the dot product $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2}$ is maximized when words i and j appear together in a window of size c. In the co-occurrence network the link indicates the words are adjacent at least once.
- In the weighted version of the networks, the weight a_{ij} of the edges reflects the likelihood of observing word i and j near each other: for the embedding network the weight a_{ij} is given by the dot product $\mathbf{w}_j^{M_1} \cdot \mathbf{w}_i^{M_2}$ which is increased each time words i and j appear in the same context window. As for the co-occurence network, a_{ij} is the number of times each word words i and j appear consecutively.

3.3.1 The Embedding Network from Shuffled Text

The two networks have touching points in terms of the structural features they represent, but an important difference between them is the contrasting way they behave when constructed from shuffled text: While the macroscopic properties of the co-occurrence network remain virtually unchanged, the embedding network is significantly changed, as the network contains virtually zero edges.

This fact becomes evident by retrieving the result of section 2.6.3 which shows that the mass does not exist in shuffled text. In the embedding network, the edge count is equal to the word mass count, that is, if in shuffled text all words have mass zero, then in the corresponding embedding network there will be no edges connecting the nodes.

This network therefore makes it possible to model the text in the form of a complex unsupervised network without having the macroscopic properties almost completely defined by the word frequencies and without bypassing text's structure.

3.4 Constructing the Embedding Network

The following code snippet illustrates the construction of the directed and weighted embedding network in Python 3 programming language. The matrices `M1` and `M2` (lines 5 and 6) result from a trained word2vec the same way as in section 2.5.2. The network is represented as an instance of `nx.DiGraph()` (or `nx.Graph()` for undirected edges) from the `networkx` library - a python library for building and analyzing graphs and networks.

```
1  import networkx as nx
2  import numpy as np
3
4  top_n_words=40000
5  M1=Model.wv.syn0[0:top_n_words]
6  M2=Model.syn1neg[0:top_n_words]
7
8  #nx.Grpah() indtead of nx.DiGraph() for undirected network
9  emb_network = nx.DiGraph()
10
11 emb_network.add_nodes_from(Model.wv.index2word[0:top_n_words])
12
13 for word in range(top_n_words):
14
15     connections_vector=np.dot(M2[word],M1.T)>0
16
17     for i in range(len(connections_vector)):
18
19         if(connections_vector[i]):
20
21             Weight = np.dot(M2[word],M1[i])
22
23             #remove "weight=Weight" for unweighted network
24             emb_network.add_edge(Model.wv.index2word[word],
25                         Model.wv.index2word[i], weight=Weight)
```

The comments on lines 8 and 23 are the adaptations to make the network undirected and unweighted, respectively.

3.5 Results

Following the results of Chapter 2 where we found that the CBOW word2vec architecture was more adequate for the measuring of the mass, we used only this architecture for the results drawn in this section. The same pre-processing and 14MWC were also used for all trials as well as the procedure of considering only the 40000 most frequent words in the corpus.

3.5.1 Retrieving the Definition of Mass

Fig. 3.4 shows the in and out-degree distributions of the embedding network. Note that the plot 3.4A of to the out-degree k^{out} matches the plot 2.6B regarding the mass distribution. This would be expected since both share a word2vec models with the same parameters and input corpus. In other words, as mentioned in section 3.3, the out-degree of the embeddings network retrieves the definition of mass.

It is observed that the in-degree k^{in}, like k^{out}, has a distribution in an apparent double power-law regime wich is a feature also seen on co-occurence networks Choudhury et al. [75]. The power-law fits in both distributions reveal different exponents in the regimes indicating that

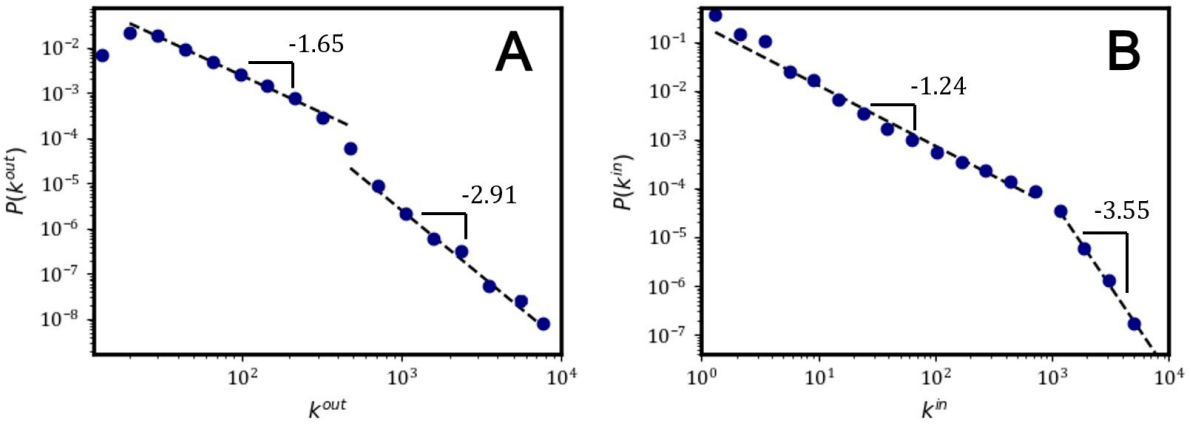

Figure 3.4: (A) In-degree k^{out} distribution for the directed embedding network. (B) Out-degree k^{in} distribution for the directed embedding network. The embedding network is assembled on a CBOW word2vec model with dimensionality $N = 200$, window size $c = 5$. Dashed lines are power-law fits with the respective exponent.

the quantities are not distributed in the same way. Furthermore, the exponent in the second regime is similar to that of the Barabási–Albert model ($\gamma = -3$) [54]. The Barabási–Albert model leads to scale-free distributions using the rule of preferential attachment. The rule states that new nodes in the network are preferentially attached to an existing node with a probability proportional to the degree of such a node.

Naturally any network obtained is dependent on the hyperparameters used in the word2vec model. Fig 3.5A shows how the window size parameter c influences the total number of edges of the embedding network (equivalent to the total mass of the system in Γ). The expected tendency is that increasing the context window increases the total number of edges since increasing c means more training examples that enable the arising of links. The trend holds with the exception that the number of edges at $c = 2$ is lower than at $c = 1$. The variation of the number of edges as a function of dimensionality is shown in Fig. 3.5B. It turns out that this hyperparameter has little influence on the size of the network, at least when framed within the recommended values $N \in [100, 500]$ [28].

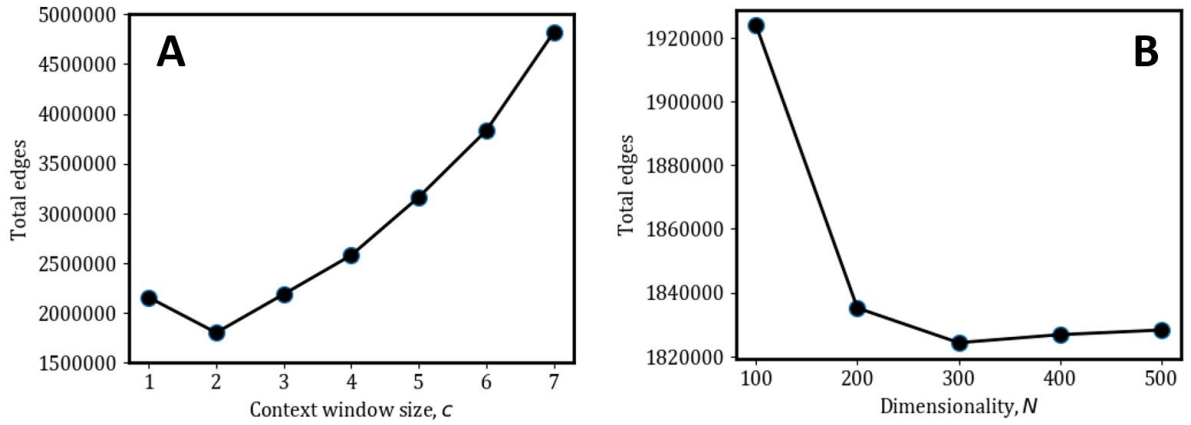

Figure 3.5: (A) Total edges of the embedding network as a function of the window size, c. (B) Total edges of the embedding network as a function of the dimensionality of the embedding space, N.

The results in Fig. 3.5 therefore show that some word2vec hyperparameters have more impact on the final network than others but the important to recognize that the different hyperparameter sets result in different networks and therefore, for a correct comparison of embedding networks, the construction must be done with the same set of hyperparameters.

3.5.2 Macroscopic Properties of the Embedding Network

We measured some of the macroscopic properties for embedding networks and a co-occurrence network. The results are shown in Table 3.2. First, it should be noted that the embedding networks exhibit low L and high C (compared to a random Erdős–Rényi network), which is reflected in the high small-world coefficient, S^{Δ}. The small world behavior was already known for co-occurrence networks [67] and it is now verified that, according to the criterion of $S^{\Delta} > 1$, the embedding networks reproduce the same behavior.

It should also be noted that a comparison can be established between the embedding network with $c = 1$ and the co-occurrence network since, in the second, adjacent words are connected and having $c = 1$ for the embedding network means that only words adjacent to the central word are considered for the training examples.

	Embedding Network, c=5	**Embedding Network, c=1**	**Co-occurrence Network, N=40000**
$\langle k \rangle$ **(undirected network)**	140,8	81,25	102,2
$\langle k^{out} \rangle$ **(directed network)***	79,53	54,03	56,02
Clustering Coefficient**, C	0,48	0,19	0,54
Average Path Length**, L	1,99	2,70	2,14
Small World Coefficient**, S^{Δ}	11,4	6,55	17,7

Table 3.2: Network properties of two embedding networks with window sizes $c = 5$ and $c = 1$ and a co-occurrence network. All networks have 40000 nodes corresponding to the 40000 most frequent words in the 14MWC. * $\langle k^{in} \rangle$ is strictly equal to $\langle k^{out} \rangle$. ** Values for the undirected network.

Comparing the edges existing in both networks, it is verified, firstly, that the values k and k^{in} are relatively similar and that 52% of the edges existing in the embedding network are found in the equivalent co-occurrence network. The percentage is enhanced if we think that, in both networks, the total edges corresponds to less than 0.15% of the $\approx 40000^2$ of possible connections.

The similarity between the network topologies supports the claim in section 3.3 that, despite major differences in construction, co-occurrence and embedding networks have touching points in the linguistic features each represents.

3.5.3 Microscopic Analysis - Nodes and Edges

Carrying out a microscopic analysis of the network implies looking individually at the components: nodes and edges. In this type of systems with a high number of elements, the individual

analysis is generally less relevant for its description. However, in the case of our language networks, we can look at the words with highest degree and at the strongest edges.

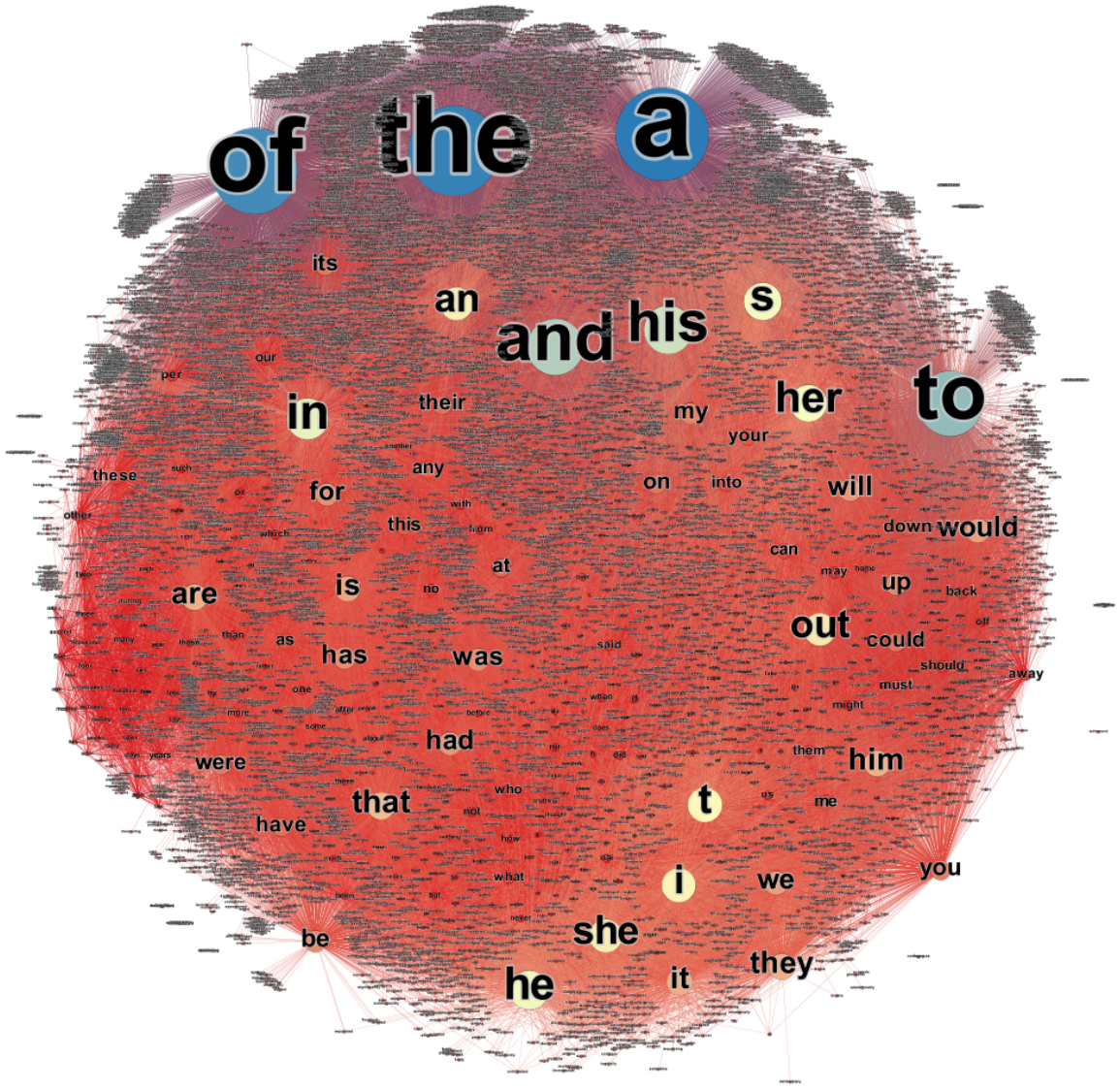

Figure 3.6: Visualization of the embedding undirected network composed of the 10000 most frequent words of the 14MWC. Dimensionality $N = 200$ and window size $c = 5$ used for the CBOW word2vec model. Node size and color gradient reflect the node's degree.

Fig.3.6 is a view of an embedding network in a force-directed layout where the size of the node - and its label - is proportional to the node's degree. We note that words that are clearly legible in the figure, i.e. those with the highest degree, are of low specificity ("the", "a", "of", "to", "and", "his", "in", etc.). This observation and the respective explanation are overall equivalent as for the words with greater mass that were made in the 2.6 results section of Chapter 1, although, now, the concept of network and connections between words makes the concept of mass more intuitive as it provides a way of visualizing it. In short, the specificity of a word is inversely related to its frequency in the text and, generally, words more often tend to share context with a greater amount of other words, which leads to them forming more links. The same is essentially

true for the co-occurrence network, so that, in this network, the words that show highest degree are essentially the same regardless of the text sample [67].

The strongest edges of the embedding network and the co-occurrence network are listed in table 3.3. For the co-occurrence network, such edges are the most frequent word bigrams in the corpus. Which are known to be, in the English language, pairs of prepositions followed by a determiner ("of the", "in the", "to the", etc.) [76, 77]. None of the same edges appear in the five strongest of the embedding network, which means that the high frequency of the bigrams alone is not decisive for obtaining strong edges. For an edge to be strong it is important that the pair is frequent but also that its elements are as "exclusive" as possible to each other. That is, an edge like "hong"→"kong" is predictably strong because either word is hardly used without the other. This causes the product $\mathbf{w}_{hong}^{M_1} \cdot \mathbf{w}_{kong}^{M_2}$ that defines the edge's weight to be almost exclusively increased with each training example. On the other hand, the more generic bigrams like "of"→"the", "in"→"the" or "to"→"the", although highly frequent, are formed by generic words whose vector representations are constantly updated in multiple directions rather than just one. That is, each time the "of"→"the" edge is strengthened, the "in"→"the" and "to"→"the" edges are weakened, as they also contain the word "the".

	Top 5 Strongest Edges	
	Embedding Network	**Co-occurrence Network**
1	"united"→"states"	"of"→"the"
2	"hong"→"kong"	"in"→"the"
3	"don"→"t"*	"to"→"the"
4	"kong"→"hong"	"on"→"the"
5	"mart"→"wal"	"for"→"the"

Table 3.3: Top five strongest edges for the directed and weighted embedding and co-occurrence networks. Dimensionality $N = 200$ and window size $c = 5$ used for the CBOW word2vec model. *The corpus pre-processing removes non alphabetical characters and so the contraction "don't" is assumed as two words "don" and "t".

It is also noted that, for the embedding network, the direction of the edges is not necessarily indicative of the order in which the words appear in the text: "kong"→"hong" and "hong"→"kong" are both strong edges even though they come from the same collocation "hong kong". "mart"→"wal" is also a strong edge whose direction does not reflect the collocation "wal mart". Such happens because the word2vec model uses, for training, a context formed by c words to the left and right of a central word without distinction (see Fig. 2.3).

In the future, by changing the word2vec training process so that training pairs include only words that appear to the right (or only to the left) of the central word, the edge direction would give a more representative notion of the sequence of words in the text as happens with the co-occurrence network. However, such a change could also have a negative impact on training effectiveness as the distributional semantics hypothesis, arguably word2vec's major principle, is clear about the importance of considering the surrounding words to retrieve a sense of meaning.

Thoughts on further work and application of the embedding network are left for the final chapter Discussion and Conclusion.

Chapter 4

Mass as Measure of Information

In this chapter, the definitions of the terms "complexity" and "information" are discussed and that serves as a motto to introduce a measure for the information present in text that corresponds to a more intuitive notion of "information".

4.1 Complexity and Information: Two Subjective Concepts

So far we have identified in linguistic systems some of the key features exhibited by complex systems such as emergence (of what we identify as "meaning"), spontaneous order (Zipf's law and other regularities) or interdependent elements (modeled by complex networks). In chapter 1 we also mentioned that complexity, being identifiable, is not easily measurable. Despite several proposals to define the quantity (or even the concept) [78, 79], an eventual unified definition of complexity was never unanimous [80, 81].

The difficulty relates to the very subjectivity of the concept of complexity. Definitions are usually criticized for one of two reasons. (1) They are not universal as they depend on subjective decisions of an external element in relation to what is considered "complex" (2) They equate "complexity" with "randomness" becoming essentially measures of the entropy (disorder) of the system. A good visual example are the patterns in Fig. 4.1. The pattern that we would intuitively identify as the most complex would be pattern B. This is however not the one with the lowest (the regular pattern A) nor highest (the random tile of pattern C) complexity according to the most common complexity definitions.

Gell-Mann et al. [82] described a concept of "complexity" that *"corresponds to our intuitive notion of complexity"* which he called *effective complexity* (EC). In a non-technical language, EC can be described as being *the length of a concise description of a set of regularities in a system* [83]. The definition of EC implies minimums for highly regular systems as well as highly random systems and a maximum of complexity for intermediate systems like pattern B.

Gell-Mann et al. claimed that EC was context dependent and subjective. To quote the very "*It (EC) depends on the coarse graining (level of detail) at which the entity is described, the language used to describe it, the previous knowledge and understanding that are assumed, and, of course, the nature of the distinction made between regularity and randomness*" [82]. Although sometimes criticized [84], EC intended to make the concept of "complexity" more intuitive by complementing the earlier definitions of Solomonoff-Kolmogorov-Chaitin[a].

Kolmogorov complexity, K, also called, among others, algorithmic complexity or algorithmic entropy, is a measure of complexity given by the extension of the shorter computer program that produces a given object as an output in a fixed universal description language (like a programming language) [85]. Take a classic example with two 36-character strings:

(1) `"ABABABABABABABABABABABABABABABABAB"`

(2) `"CG4OUO1NBAZERTEW3U8YISNWYAK5OYZQTV6G"`

In Python 3 programming language, string 1 can be outputed with `Print("AB" * 18)`. String 2, on the other hand, does not have an obvious simpler description than printing the string itself, i.e., `Print("CG4OUO1NBAZERTEW3U8YISNWYAK5OYZQTV6G")`. Any pattern or regularity lowers the complexity. This is because a pattern constitutes redundancy: it enables one portion of

[a] Three authors are credited for independently describing the same theorem [85]. In chronological order: R.J. Solomonoff [86, 87, 88] (Cambridge, Massachusetts, USA); A.N. Kolmogorov [89, 90] (Moscow, Russia); and G.J. Chaitin [91, 92] (New York, USA). The works gave rise to the quantity that become well entrenched and commonly understood as *Kolmogorov complexity*.

the string to be recovered from another, allowing a more concise description. In theory, K is applicable to any object or system. Again in the patterns example, pattern C is the one that will have the highest K since it is a random pattern. Pattern A, on the other hand, has a low K since is highly regular.

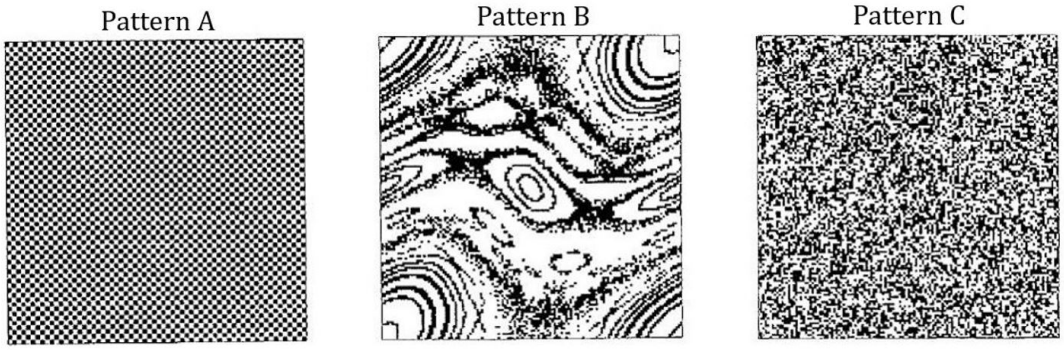

Figure 4.1: Three patterns with different Kolmogorov complexities, K. The pattern one intuitively identifies as more complex (or interesting), B, is neither the one with the lowest K, A, nor the one with the highest, C. Adapted from [80].

Text A	Text B	Text C
the the the the	It is, in fact, nothing	exuberant geese
the....of of of of of	short of a miracle	chase throne the
of....and and and and	that the modern	volleyball .....
and....to to to to to....	methods of	..... impartial
.....a a a a a..... and	instruction have not	adhesive brash
and and and.....	yet entirely strangled	happy copy dusty zip
.....theory theory.....	the	random yellow in
einstein einstein.....	.....	relativity.....

Figure 4.2: An extension to the analogy of Fig. 4.1 with text samples. Text A is a discourse ordered according to word frequencies (highly regular); Text B is actual discourse; Text C are random selected words.

Gell-Mann et al. proposed dividing the Kolmogorov complexity that describes an object in two terms: one for regularities and the other for features considered random or accidental. The first term is then the effective complexity - the minimum description length of the regularities of the entity [82]. Note that both K and EC are quantities not computable by a formula (but can be approximated by upper and lower bounds). On the other hand, Shannon entropy is a quantitative measure of "randomness" or "uncertainty" associated with the outcomes of a random variable. It is closely related to statistical thermodynamics of Boltzmann and Gibbs and the physical concept of entropy that is a measure of disorder [93, 94]. The Shannon entropy, H_s, depends only on the probabilities of occurrence of V possible outcomes of a random variable: $p_i (i = 1, 2, ..., V)$

$$H_s = -\sum p_i \log_2 p_i \tag{4.1}$$

Shannon entropy takes each event as random without considering possible regularities. In the case that both patterns A and C in Fig. 4.1 have the same number of white and black dots (say

half of each), in both patterns the entropy associated with each dot is the same

$$H_s = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1 \tag{4.2}$$

That is, all permutations of the elements of the system are therefore equivalent in terms of H_s.

Having established the main differences between the quantities[b] EC, K and H_s, we can now turn attention to linguistic systems by starting extending the analogy shown in Fig. 4.1 to texts as being sequential collections of words (Fig. 4.2). Texts A and B have the same words and thus the same entropy H_s. Only that B is actual discourse (with "meaning") and in A the words are organized in descending order of frequency. Both have the same entropy but the organized structure of text A makes it the one with the lowest K. Text C, being a collection of random words, is the one with the highest entropy (each word has the same $1/V$ probability of appearing next) and the highest K (the one with the least regularities).

Here we take the opportunity to introduce the second subjective concept of *information*. The concept usually accompanies developments concerning measures of entropy and complexity. In fact, H and K are two of the most common measures of information since the amount of information usually corresponds to its "unpredictability". In such a way that it is stipulated that an event of probability 100% is perfectly unsurprising and yields no information.

From physics perspective, entropy H concerns the system as a ensemble, i.e., the probabilistic formalism where the set of microstates defines the set of probabilities then used to calculate H (whether it be the Shannon, Gibbs or von Neumann entropy for quantum systems). On the other hand, K does not use the ensemble strategy and can be applied to the individual microstates of the system. Briefly explained by Zurek [94] "(about K) *We shall imagine that the microstate of the system is known, and our task is to communicate it to someone else or to record it in some reproducible fashion*".

The debate over the applicability of H or K as measures of complexity/information exists but both are established quantities that are undoubtedly relevant depending primarily on the context in which they are used.

To Gell-Mann et al. [83], EC also has a place as a measure of information that illustrates a more intuitive concept and states that both highly random and highly regular objects carry little information. He posed the question using a linguistic example: *How could Shakespear's works, which show some degree of regularity, have a smaller information content than a string of random words of the same length?* [95]. Information theory then seemed to have a gap in the lack of relevance given to systems that exhibited complex regularities and emergent structures. A gap that could be filled by the effective complexity.

In another EC description, the authors compare the EC to the complexity of the program responsible for generating an object (not counting the data fed into that program) [82]. Let one consider flipping a fair coin 1000 times while recording a 0 for heads and 1 for tails. One would end up with a random-looking string of bits such as `01101001....0110`. The precise

[b] In the context of this Thesis we feel sufficient, and even more appropriate, to introduce the concepts of Shannon's Entropy, Kolmogorov's complexity and effective complexity in a conceptual rather than a formal way. *An Introduction to Kolmogorov's Complexity and Its Applications* [85] provides full formal descriptions of K and comparisons with H_s. The literature on EC includes essentially the works of Gell-mann and Loyd [82, 83, 95]

string would probably take almost 1000 bits to describe, yet one can succinctly describe the process as "flip a coin 1000 times", but that doesn't encode the exact string of bits that running this process generates. This exemplifies a low EC process. Analogously, an algorithm is easily programmed to create a random text - like text C in Fig. 4.2 - but an algorithm fully capable of elaborating speech in a human-like way is still not conceivable (let alone one that writes a Shakespearean play).

4.2 Towards *Effective Information*

Gell-Mann et al.'s definition of effective complexity fits the way in which, in this work, we approach the linguistic system. That is, a system where the real information/complexity comes from a structure that is part regular and part random. And only in the presence of this structure (synthesis rules, semantic relations, hierarchical relations, etc.) is the maximum information/complexity reached.

In that line, we propose a measure for the information content of a natural language text that reflects a more intuitive idea of the term "information". To such quantity we should call *effective information*, and, to use Saussurean terms, it can be seen as the amount of distinct signs corresponding to distinct ideas present in the text as an information vector. In order to reflect the intuitive definition of information in text, effective information must respect three condition:

I. In text that does not respect the linguistic rules of syntax/semantics (i.e., text without meaning) the effective information must be null.

II. Any information that is repeated or redundant must not contribute to increase the effective information.

III. In the absence of structural irregularities (I) or redundancies (II) the effective information should increase with the size of the corpus.

The three conditions reflect the concept of information quite intuitively. Condition I emphasizes the importance of the text's emergent structure for the acquisition (or transmission) of information. The natural consequence of I is that texts like A and C of Fig. 4.2 have minimum values of effective information. Condition II implies that the repetition of ideas, which can occur through the repetition of phrases or other portions of text in a corpus, does not contribute more than once to effective information. This condition recovers the Kolmogorov complexity assumption that redundancy does not increase system complexity. As for condition III, it reflects the ideas, described by Saussure himself, that each word, as a basic unit of language, is equivalent to a sign, and then, under normal structural conditions, more words correspond to more signs and therefore more information.

Once again we emphasize that the way the term "information" is used here should not be confused with the connotation of the term in information theory. In that context, "information" is a measure of "uncertainty" while here the term has a perhaps more cognitive-associated dimension serving as measure that quantifies the amount of "ideas" expressed in text.

4.3 Mass as a Measure of Information

We first identified the role of effective information as a measure of textual information and also the three conditions on which the concept is based. We now follow by associating the proposed effective information to the mass M introduced in equation 2.12. Concretely, we propose that the total mass of the system serves as a metric of effective information since, as we shall see, its definition and the results obtained in different *corpora* (plural of *corpus*) indicate that the conditions are respected.

Let M_{w_i} be the mass of the word w_i ($i = 1, 2, ..., V$) of a text that was subjected to the transformation $\Omega \to \Gamma$ performed by word2vec. The *effective information*, Y, of the text is

$$Y = \sum_{i=0}^{V} M_{w_i} \tag{4.3}$$

That is, Y is the sum of all the individual word masses of the system in Γ.

As we saw in the results section of Chapter 2, the word's mass is null on shuffled (structurally irregular) text. Such observation is easily associated with condition I: In the absence of correctly structured text, the total mass of the system, and therefore effective information, is null.

Condition II deals with redundant or repeated information that, in text, we can think of as repetition of portions of the corpus. Weather they are sentences, entire paragraphs or simple collocations. In the word2vec training process, such repetitions mean training examples already seen by the model, which, in theory, imply minor adjustments to the word vector at each repetition and therefore will have less significant contributions to the mass. Furthermore, the repetition of portions of text increases the corpus dimension without increasing the vocabulary size. Perhaps the subject becomes clearer with an example: Let there be a 10 million word corpus formed by a single ten-word sentences repeated a million times. Even though it is indisputably extensive, the total mass of such a system represented in Γ would always be low due to a vocabulary, $V = 10$, very reduced.

Condition III is also easily associated with mass. Increasing the size of a corpus also implies increasing the vocabulary V in a way that is generally regular (the empirical regularity known as Heaps' law [96]). Moreover, with the increase of the corpus, the diversity of contexts increases, which translates into more diversified training examples. Both of these consequences of corpus increasing predictably lead to an increase in the total mass of the system.

4.3.1 Mass is Uniquely Defined

The works of Gell-Mann and Lloyd [82, 83, 95] generally missed two aspect that were pointed out by others [84, 97] as important limitation to the measure's applicability: (1) EC is not uniquely defined because the distinction between random and regular features of a given object is arbitrary and dependent of external judgment. And (2): Even assuming limitation 1 is fixed, a way to generically computing EC for any object is not provided (as with K).

The effective information Y we propose is a quantitative, generic and scalable method to the

application of EC's premises to linguistic systems. It is also unsupervised and therefore uniquely defined.

The transformation $\mathbf{M_1} : \mathbb{R}^V \to \mathbb{R}^N$ performed by word2vec works as a dimension reduction of the system that is only possible given that the system has redundancies (or regularities) when represented in $\mathbb{R}^V$. As mentioned in Chapter 2, the word2vec algorithm employs the distributional semantics hypothesis along with distributed representation to identify similarities among words and then "suppress" the redundant dimensions.

This is only possible as the text has a non-random structure. Stereotyped expressions, hierarchical relations and collocations are all forms of regularities that reduce the uncertainty of the system. In the case that text is random, the redundancies inherent to the linguistic structure are suppressed and therefore the dimensionality reduction is not effective - a result that has already been obtained in the section 2.6.3 and which we retrieve ahead.

4.4 Results

In order to verify if conditions I, II and III were satisfied by the definition of Y, some tests were carried out involving the measurement of the total corpus mass with different characteristics. As for the embedding model and mass measurement, the methodology used is analogous to that described in the 2.5 section. The trials were carried out using CBOW architecture, window size $c = 5$ and only considering the 40000 most frequent words as explained in the section 2.5.2 (unless otherwise indicated). All the remaining word2vec hyperparameters are also constant.

4.4.1 Effective Information Decreases as Text is Shuffled

To test condition I, Y was computed for a corpus that is progressively shuffled. The shuffling process gradually suppresses the structure and underlying ideas in the text. A shuffle step consists of randomly selecting two words from the corpus and swap their position. 1.5×10^7 shuffle step were progressively applied to the 14MWC (see script in Appendix D). The results are shown in Fig. 4.3.

For both dimension tested, $N = \{100, 400\}$, the effective information decreases with the shuffling and eventually reaches a value that is virtually $Y = 0$ when the entire text is shuffled. This shows that condition I is respected, but also that the decrease in information is gradual. Balasubrahmanyan et al. [98] wrote that the emergence of "meaning" resembles a phase transition of a physical system. This statement is debatable because it would mean that a corpus would function as a ensemble that fits into one of two possible macrostates: "with meaning" or "without meaning". In our view, "meaning" comes from the microstates within the system. It somehow arises from every word, phrase, sentence, paragraph, etc. From the wide range of possible microstates, only a small fraction is considered "meaningful". For example, there is the possibility that a sentence has "meaning" (i.e. to be informative) and the sentence that follows does not. Or that a portion of partially shuffled text is considered "partially informative." The gradual decrease in Fig. 4.3 reflects this judgment: The shuffling gradually changes the microstates from "informative" to "non informative" leading Y to strictly decrease throughout.

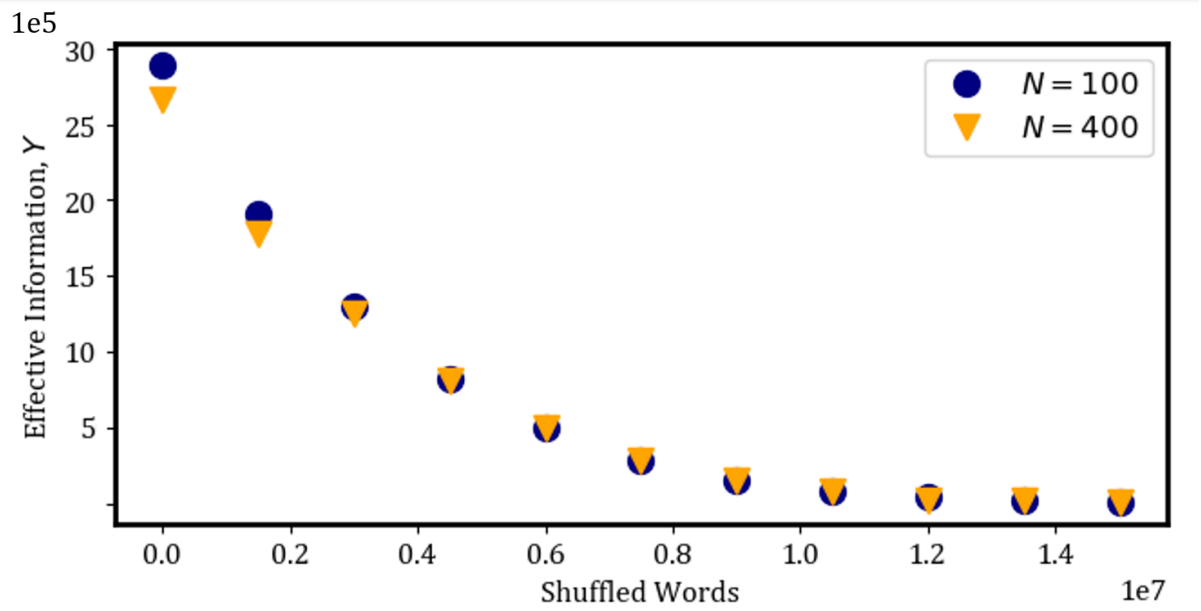

Figure 4.3: Variation of effective information Y with gradual word shuffling. The 14MWC is progressively shuffled using the `randomize_sentences` method in Appendix D. For both dimensionalities tested, $N = \{100, 400\}$, the effective information decreases with the shuffling and eventually reaches a value that is virtually $Y = 0$ when the entire text is shuffled. The results go accordingly to condition I of effective information.

4.4.2 Effective Information Increases with Corpus Size

We tested the variation of Y with corpus size. If condition III is respected, Y should increase with the size of the corpus as more text contains more information. Different fractions of the 14MWC were used to create 7 corpora of different dimensions (from 3.8 million up to 14 million words). The plot of Fig. 4.4 shows that Y is monotonically increasing with the corpus size as predicted. There is a slight flattening of the curve of Y values for larger corpus sizes which suggests that the underlying relationship is not linear. This can be explained if we think that in a corpus that is growing, it is inevitable that ideas/information will be repeated at an increasing rate (Heaps' Law [96]), and, according to condition II, repeated information has a smaller contribution to Y than new information (see Fig. 4.5 and Section 4.4.3).

4.4.3 Effective Information On Redundant *Corpora*

The last set of tests is related to condition II and the effect of redundant information in Y. The corpora with redundant information contains repeated portions of the 14MWC (one or more times).

Fig. 4.5A shows Y in corpora formed by multiple sequential repetitions of the 14MWC (1 to 4 repetitions). The four corpora all have the same vocabulary and the same sentences, only the number of times each one is repeated varies. According to condition II, Y should be constant for the four trials given that no new information is added despite multiple corpus sizes. The training examples and contexts are also the same. However, the repetitions cause the model to continue to adjust the embedding vectors even with repeated information causing Y to increase with the repetitions of the corpus in Fig. 4.5A. This happens because the system is not in what

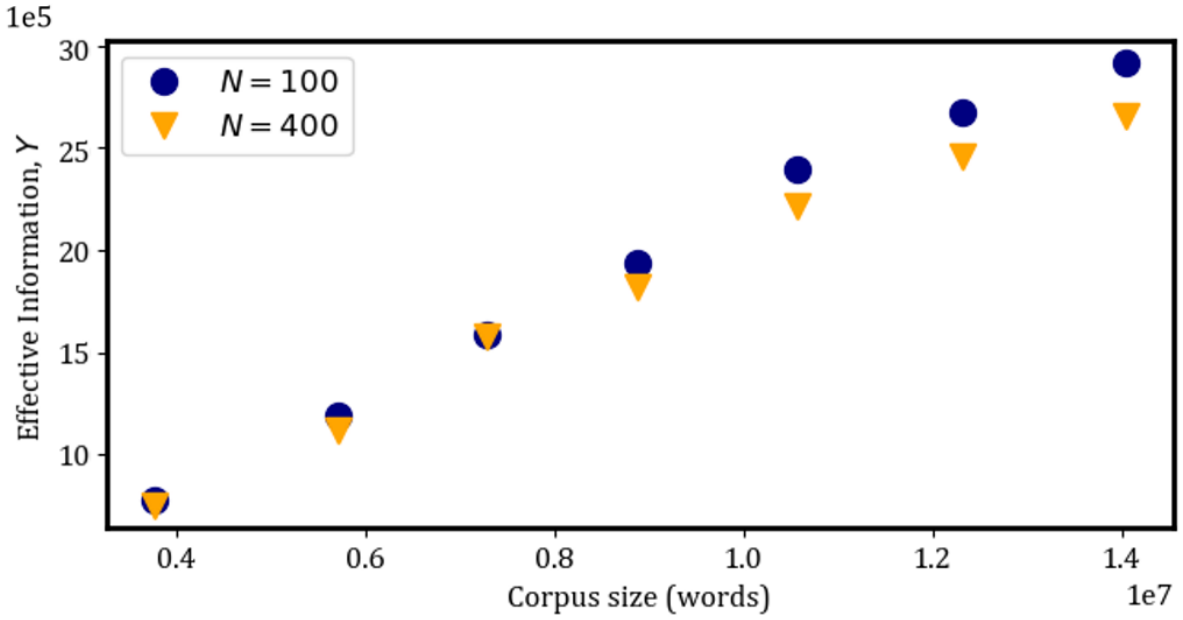

Figure 4.4: Effective information Y as a function of corpus size. It can be seen that effective information increases with corpus size which respects the condition III of effective information.

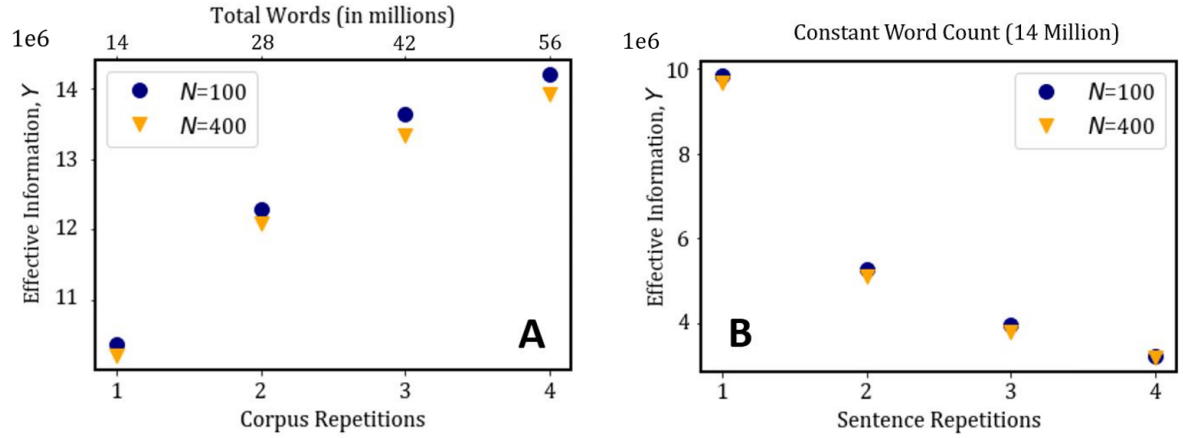

Figure 4.5: (A) Effective information Y on corpus with repeated information. The horizontal axis indicates the number of times the corpus of 14 million words was repeated in the trial. All the trials have the same sentences and therefore the same information. Only the number of times each sentence is repeated varies. According to condition II, Y should remain constant as repeated information should not contribute to increase it. However, Y constantly increases even if at a lower rate than in Fig. 4.4. (B) Effective information Y variation on constant size corpus with repeated information. The horizontal axis gives the number of times each sentence is repeated on a constant size 14 million corpus. Y decreases with sentence repetition which goes accordingly to condition II. Note: for the purpose of mass computation, all the words in each corpus were considered (and not just the 40000 most frequent ones) to show the effects of changing the size of the vocabulary.

would be considered a state of equilibrium in which the training process would no longer change matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ significantly. Remember that word2vec functions as a RBM and therefore it should be possible to reach the point where the probability distribution of global states has converged when system reaches a state of equilibrium.

The fact that the system is not in equilibrium makes direct verification of condition II somewhat inconclusive. We also tested redundant information on constant size corpora. 4.5B shows the

computed Y for multiple corpora all composed of 14 million words. The number of times each sentence is repeated varies between 1 and 4 times. As Y decreases along the horizontal axis, we see that Y doesn't just reflect the size of the corpus. The results from 4.5B clearly show an effect that follows from condition II: If two corpora have the same size, the one with more redundant information will have the lowest Y.

In the future, it is important to test more effectively the impact of repeated information on Y and then test whether condition II is respected by the definition of eq. 5.2. It also must be checked if the system reaches a state that can be considered as an equilibrium. Also relevant is the analysis of different corpora. In this work we restricted ourselves to using the 14MWC as proof of concept of the methods. In the future, comparing effective information from different texts - different genres and different languages - will allow for more robust conclusions to be drawn about the applicability of the concept.

In the final Discussion and Conclusion chapter, we conclude the discussion about effective information and point out applications and future work.

Chapter 5

Discussion and Conclusion

A last review over this work's main contribution and some final remarks on further work, possible applications and the future of quantitative linguistics.

At the beginning of this thesis, we introduced the framework where natural languages are modeled as complex systems. Even if such an approach was not entirely new, our work differed by representing the system in the space of reduced dimension by means of neural-based word embedding method. The so called *embedding space* was equipped with a similarity metric between elements and a new property of attractiveness: the *mass*. The choice of the term "mass" arises from the physical property of bodies that determines the force of gravitational attraction. In our *universe of words*, mass is a property of *attractiveness to use* in an N-dimensional system that is just made of the words and the interactions between them. It is on this property that the innovations proposed throughout chapters 2, 3 and 4 are based. Here we summarize the key messages from this work:

- Natural languages displays many of the features normally associated with complex systems such as emergent behavior. We argue that approaches which ignore the emergent structure do not produce fully adequate models to describe the system.
- The word embedding model word2vec acts on text performing a dimension reduction and representing each word in a vector space where a semantic similarity metric can be defined. In the same vector space, the property mass M was defined (eq. 2.12).
- M is a property that retrieves the *attraction to use* given by word frequencies. The results show that mass and frequency are positively correlated and both are distributed according to power laws.
- Unlike frequency, the mass is changed with a shuffling of the text. The fact that mass does not exist in shuffled text (virtually $M_{w_i} = 0$ for all V word that compose the vocabulary) shows that the quantity does not bypass text emergent structure.
- In Chapter 3, we introduced a new form of unsupervised linguistic networks to which we call *embedding network*. The embedding network is based on the definition of M_{w_i}. The proposed network shows scale-free and small-world properties.
- We have shown that the embedding network is the only unsupervised linguistic network whose macroscopic properties are significantly altered on shuffled text.
- In chapter 4 we showed that the total mass of the system can be treated as a measure of information/complexity in line with the ideas of effective complexity by Gell-Mann and Lloyd. We further define the quantity effective information, Y, based on three conditions that reflect the intuitive concept of "information".

Our aim for this work was never to draw major conclusions in the scope of linguistics but rather to review and present innovative ways of targeting the system's complexity. Essentially, we hope to have created some foundations so that the ideas proposed in this work can be useful when applied the field of quantitative linguistics or natural language processing tasks.

In this work, the same 14 million word corpus served as input data for all the tests performed. We purposely avoided testing different texts to avoid accounting issues related to the diversity of natural languages and literary genres. The proposed methods are unsupervised and generic

and therefore easily applicable to any corpus. An important step in the future will be to gather diversified texts (of different genres and languages) and compare their macroscopic properties according to this work's framework.

The study of characteristic exponents[a] γ of the mass probability distribution

$$P(M) \propto \frac{1}{M^{\gamma}} \tag{5.1}$$

may function as an unsupervised form of text classification if it is found to be similar for texts of the same genre or language. The same application has already been suggested using word frequencies [99]. Further work should also include evaluating the influence of word2vec hyperparameters on γ. The fact that mass is a property sensitive to "meaningful text" suggests that it can act as a criterion for identifying structural anomalies or artificially generated text.

In chapter 3 we explained how the definition of mass can be extended to serve as a criterion for building an *embedding network*. This new form of unsupervised linguistic network is also sensitive to "meaningful text" and therefore represents advantages in terms of system description comparatively to the co-occurrence network. Despite some interesting results, it is not guaranteed that the complex network approach is the most suitable for describing languages or modeling the mental lexicon [100]. Yet, different linguistic networks are certainly useful in specific studies or tasks. Based on previous works, it is our understanding that the embedding network shows potential for application in tasks such as language translation [101, 102], identification of literary movements [103], or perform cluster-based classification of natural languages [64]. All these tasks make use of the macroscopic properties of the network.

The term "complex" is found numerous times and in diverse contexts throughout this work[b]. In chapter 4 we saw that *complexity* and *information* are two concepts that sometimes blend and that are at least subjective. The idea that *effective information,* Y, defined as the sum of all masses of the system

$$Y = \sum_{i=0}^{V} M_{w_i} \tag{5.2}$$

represents a more intuitive form of "information" present in the text is in line with Gell-Mann and Loyd's visions of effective complexity. More importantly, the proposed method is unsupervised so it does not rely on subjective opinions to distinguish regularity from random features. Although, in theory, the definition of Y is adequate as a measure of effective information, we stress that is strictly necessary to fully understand whether condition II is in fact respected or not (or only partially).

Our metric makes use of the relationships between words to identify the regularities that allow the system to be fitted in a space of reduced dimension. The procedure is potentially applicable to objects other than text: A project called Signal2Vec [104] uses an architecture similar to that of word2vec to create vector embeddings of time-series. Embedding methods have also been developed for songs [105] or images [106, 107]. All these objects share the fact that they are non-random sequences of basic units: pictures are formed by pixels, songs by musical notes, time-series by digits and text is a sequence of words. Assuming that each of the sequences has

[a] assuming that the distribution is fitted by more than one power law regime

[b] 39 to be precise!

regularities, the object can be embedded in a space with fewer dimensions than the number of existing basic units and, with the necessary adjustments, there is no reason not to adapt Y to any of these systems.

Nevertheless, we acknowledge that, since the definition of mass is such a central matter of this work, further work should be done to clarify the influence (and limitations) of our definition in terms of architecture (Skip-Gram and CBOW) and hyperparameters of the word2vec model as well as size of the training corpus. For the purposes of this work we intended the methods used to be as much unsupervised as possible which meant that very little pre-processing was applied to the corpus. In the future, we must find if there are any corpus pre-processing techniques (stop-word removal, sub-sampling of frequent word among others) that can benefit overall performance and results.

Final Remarks

Like many other interesting systems, once you start studying language you end up finding layer after layer of complexity when searching for the ultimate theory to describe it. The challenges sure make one doubt about the chances of one day achieving a sophisticated form of AI that interprets and communicates as efficiently as humans. It must not be forgotten that language was not created in a centralized way. The rules, the meanings, the expressions were established by recurrent use of the speakers. The linguistic system was self-organized through the input of its users and evolution is constant. The idea is well summarized by Borges' quote found at the beginning of this thesis: *"All language is a set of symbols whose use among its speakers assumes a share past."*

We definitely believe that this work presents valid tools that can be useful in the field of quantitative linguistics and we encourage anyone to employ and help develop the ideas we proposed.

On a final comment, Richard Feynman had a joke about *a posteriori* conclusion, reasoning the cause from the outcome. He would say:

"You know, the most amazing thing happened to me tonight. I saw a car with the licence plate ARW 357. Can you imagine? Of all the millions of licence plates in the state, what was the chance that I would see that particular one tonight? Amazing!"[108]

His point is that it is easy to make any result seem extraordinary if treated as fateful or very unlikely. The problem is somewhat transverse to the field of complex systems. Authors sometimes find it difficult to look impartially at their own work, which can lead to exaggerated conclusions being drawn. Perhaps this very work is not entirely free from this problem either. We must be aware that in some matters we are still only scratching the surface in terms of acquired knowledge. For that reason, experimentation, debate of ideas and coordinated multidisciplinary efforts are essential for advances in an area as challenging as complex systems.

Bibliography

[1] Eörs Szathmáry. "Evolution of Language as One of the Major Evolutionary Transitions". In: *Evolution of Communication and Language in Embodied Agents* (Jan. 2010), pp. 37–. DOI: 10.1007/978-3-642-01250-1_3.

[2] Marc D. Hauser et al. "The mystery of language evolution". In: *Frontiers in Psychology* 5 (2014), p. 401. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2014.00401. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2014.00401.

[3] Beth Sagar-Fenton McNeill and Lizzy. *How many words do you need to speak a language?* June 2018. URL: https://www.bbc.com/news/world-44569277.

[4] William Pickering. "Natural languages as complex adaptive systems". In: *Estudos Linguísticos (São Paulo. 1978)* 45 (Nov. 2016), p. 180. DOI: 10.21165/el.v45i1.787.

[5] Alexander Andrason. "Language Complexity: An Insight from Complex-System Theory." In: *International Journal of Language and Linguistics* 2 (2014). URL: http://www.sciencepublishinggroup.com/journal/paperinfo.aspx?journalid=501%5C&doi=10.11648/j.ijll.20140202.15.

[6] A. Baronchelli et al. "Complex systems approach to the emergence of language". In: *Language, Evolution and the Brain*. Ed. by J. W. Minett Wang and W. S-Y. 2007.

[7] Paul Hopper. "Emergent Grammar". In: *Berkeley Linguistics Society* 13 (Sept. 1987), pp. 139–157. DOI: 10.3765/bls.v13i0.1834.

[8] R. Lopez-Pena and R. Capovilla. "Complex System and Binary". In: *Springer* (1995).

[9] Constantino Tsallis. "43 Visions for Complexity". In: *World Scientific Publishing* (2017), pp. 75–76. DOI: 10.1142/9789813206854_0038.

[10] Herbert A. Simon. "On a class of skew distribution functions". In: *Biometrika* 42.3–4 (1955), pp. 425–440. DOI: 10.1093/biomet/42.3-4.425. eprint: http://biomet.oxfordjournals.org/content/42/3-4/425.full.pdf+html.

[11] B. Mandelbrot. "A note on a class of skew distribution functions: analysis and critique of a paper by H. A. Simon". In: *Information and Control* 2 (1959), pp. 90–99.

[12] Eduardo G. Altmann and Martin Gerlach. "Statistical Laws in Linguistics". In: *Creativity and Universality in Language* (2016), pp. 7–26. ISSN: 2195-1942. DOI: 10.1007/978-3-319-24403-7_2. URL: http://dx.doi.org/10.1007/978-3-319-24403-7_2.

[13] Martin Gerlach and Eduardo G Altmann. "Scaling laws and fluctuations in the statistics of word frequencies". In: *New Journal of Physics* 16.11 (Nov. 2014), p. 113010. DOI: 10.1088/1367-2630/16/11/113010. URL: https://doi.org/10.1088/1367-2630/16/11/113010.

[14] G. Herdan. *Type-token Mathematics*. Janua linguarum, Studia memoriae Nicolai van Wijk dedicata. Series maior, 4. Mouton, 1960. URL: https://books.google.pt/books?id=jJhkQwAACAAJ.

[15] George K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.

[16] Ramon Ferrer i Cancho, Chris Bentz, and Caio Seguin. "Compression and the origins of Zipf's law of abbreviation." In: *CoRR* abs/1504.04884 (2015). URL: http://dblp.uni-trier.de/db/journals/corr/corr1504.html#Ferrer-i-Cancho15.

[17] Ali Mehri and Sahar Mohammadpour Lashkari. "Power-law regularities in human language". In: *The European Physical Journal B* 89.11 (Nov. 2016), p. 241. ISSN: 1434-6036. DOI: 10.1140/epjb/e2016-70423-9. URL: https://doi.org/10.1140/epjb/e2016-70423-9.

[18] F.A. von Hayek. *Studies in philosophy, politics and economics*. 1967.

[19] Vitória Piai et al. "Direct brain recordings reveal hippocampal rhythm underpinnings of language processing". In: *Proceedings of the National Academy of Sciences* 113.40 (2016), pp. 11366–11371. ISSN: 0027-8424. DOI: 10.1073/pnas.1603312113. eprint: https://www.pnas.org/content/113/40/11366.full.pdf. URL: https://www.pnas.org/content/113/40/11366.

[20] Mo Costandi. *The brain has its own "Autofill" function for speech*. Apr. 2017. URL: https://www.scientificamerican.com/article/the-brain-has-its-own-ldquo-autofill-rdquo-function-for-speech/.

[21] F. Saussure et al. "Course in General Linguistics". In: *Journal of American Folklore* 73 (1960), p. 274.

[22] Ramon Ferrer-i-Cancho and Ricard Sole. "Zipf's law and random texts". In: *Advances in Complex Systems* 5 (Jan. 2002), pp. 1–6.

[23] Steven Piantadosi. "Zipf's word frequency law in natural language: A critical review and future directions". In: *Psychonomic bulletin and review* 21 (Mar. 2014). DOI: 10.3758/s13423-014-0585-6.

[24] Ramon Ferrer i Cancho and Ricard V. Solé. "Two Regimes in the Frequency of Words and the Origins of Complex Lexicons: Zipf's Law Revisited". In: *Journal of Quantitative Linguistics* 8.3 (2001), pp. 165–173. DOI: 10.1076/jqul.8.3.165.4101.

[25] Ruokuang Lin, Qianli D. Y. Ma, and Chunhua Bian. *Scaling laws in human speech, decreasing emergence of new words and a generalized model*. 2015. arXiv: 1412.4846 [cs.CL].

[26] Yoshua Bengio et al. "A Neural Probabilistic Language Model". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. ISSN: 1532-4435.

[27] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: (2013). URL: http://arxiv.org/abs/1301.3781.

[28] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C.J.C. Burges et al. 2013, pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

[29] François Chollet. *Deep Learning with Python*. Manning, Nov. 2017. ISBN: 9781617294433.

[30] Happiness Ugochi Dike et al. "Unsupervised Learning Based On Artificial Neural Network: A Review". In: *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. 2018, pp. 322–327. DOI: 10.1109/CBS.2018.8612259.

[31] Chris M Bishop. "Neural networks and their applications". In: *Review of scientific instruments* 65.6 (1994), pp. 1803–1832.

[32] Jianguo Xin and Mark J Embrechts. "Supervised learning with spiking neural networks". In: *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*. Vol. 3. IEEE. 2001, pp. 1772–1777.

[33] G. E. Hinton. "Boltzmann machine". In: *Scholarpedia* 2.5 (2007). revision #91076, p. 1668. DOI: 10.4249/scholarpedia.1668.

[34] Asja Fischer and Christian Igel. "An Introduction to Restricted Boltzmann Machines". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by Luis Alvarez et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 14–36. ISBN: 978-3-642-33275-3.

[35] Xin Rong. *word2vec Parameter Learning Explained*. 2016. arXiv: 1411.2738 [cs.CL].

[36] Yoav Goldberg and Omer Levy. *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. 2014. arXiv: 1402.3722 [cs.CL].

[37] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd. USA: Prentice Hall PTR, 2020. ISBN: 0130950696.

[38] Andriy Mnih and Yee Whye Teh. *A Fast and Simple Algorithm for Training Neural Probabilistic Language Models*. 2012. arXiv: 1206.6426 [cs.CL].

[39] *Word2Vec : Difference between the two Weight matrices*. 2018. URL: https://stats.stackexchange.com/questions/342174/word2vec-difference-between-the-two-weight-matrices.

[40] Emmanuelle Dusserre and Muntsa Padró. "Bigger does not mean better! We prefer specificity". In: *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*. 2017. URL: https://www.aclweb.org/anthology/W17-6908.

[41] Miguel Bernardo. "Construction of Geometries Based on Automatic Text Interpretation". MA thesis. Faculdade de Ciências e Tecnologias da Universidade NOVA de Lisboa, Nov. 2020.

[42] Shuvayanti Das et al. "Critical Dimension of Word2Vec". In: *2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*. 2019, pp. 202–206. DOI: 10.1109/IESPC.2019.8902427.

[43] Radim Řehůřek. *Gensim Word2vec embeddings*. Apr. 2021. URL: https://radimrehurek.com/gensim/models/word2vec.html.

[44] A. Akmajian et al. *Linguistics: An Introduction to Language and Communication*. Linguistics: An Introduction to Language and Communication. MIT Press, 2001. ISBN: 9780262511230. URL: https://books.google.pt/books?id=gPbQyRdnM18C.

[45] Xin Rong. "word2vec parameter learning explained". In: *arXiv preprint arXiv:1411.2738* (2014).

[46] Réka Albert and Albert-László Barabási. *Statistical mechanics of complex networks*. cite arxiv:cond-mat/0106096 Comment: 54 pages, submitted to Reviews of Modern Physics. 2001. URL: http://arxiv.org/abs/cond-mat/0106096.

[47] Albert-László Barabási, Réka Albert, and Hawoong Jeong. "Scale-free characteristics of random networks: the topology of the world-wide web". In: *Physica A: Statistical Mechanics and its Applications* 281.1 (2000), pp. 69–77. ISSN: 0378-4371. DOI: https://doi.org/10.1016/S0378-4371(00)00018-2. URL: https://www.sciencedirect.com/science/article/pii/S0378437100000182.

[48] Antonio del Sol, Hirotomo Fujihashi, and Paul O'Meara. "Topology of small-world networks of protein–protein complex structures". In: *Bioinformatics* 21.8 (Jan. 2005), pp. 1311–1315. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti167. eprint: https://academic.oup.com/bioinformatics/article-pdf/21/8/1311/691592/bti167.pdf. URL: https://doi.org/10.1093/bioinformatics/bti167.

[49] Ramasco et al. "Self-organization of collaboration networks". In: *Physical Review E* 70 (Oct. 2004), p. 036106. DOI: 10.1103/PhysRevE.70.036106.

[50] Fredrik Liljeros et al. "The web of human sexual contacts". In: *Nature* 411.6840 (June 2001), pp. 907–908. ISSN: 1476-4687. DOI: 10.1038/35082140. URL: http://dx.doi.org/10.1038/35082140.

[51] Annalisa Socievole, Floriano De Rango, and Antonio Caputo. "Wireless contacts, Facebook friendships and interests: Analysis of a multi-layer social network in an academic environment". In: *2014 IFIP Wireless Days (WD)*. 2014, pp. 1–7. DOI: 10.1109/WD.2014.7020819.

[52] Jin Cong and Haitao Liu. "Approaching human language with complex networks". In: *Physics of life reviews* 11 (Apr. 2014). DOI: 10.1016/j.plrev.2014.04.004.

[53] Antoniou Ioannis and Tsompa Eleni. *Statistical analysis of weighted networks*. 2007. arXiv: 0704.0686 [physics.soc-ph].

[54] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509. eprint: http://www.sciencemag.org/content/286/5439/509.full.pdf. URL: http://www.sciencemag.org/content/286/5439/509.abstract.

[55] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (1998), pp. 440–442. ISSN: 14764687. DOI: 10.1038/30918. URL: https://doi.org/10.1038/30918.

[56] Mark D. Humphries and Kevin Gurney. "Network 'Small-World-Ness': A Quantitative Method for Determining Canonical Network Equivalence". In: *PLOS ONE* 3.4 (Apr. 2008), pp. 1–10. DOI: 10.1371/journal.pone.0002051. URL: https://doi.org/10.1371/journal.pone.0002051.

[57] Erdös and Rényi. "On the evolution of random graphs". In: *Publication of Mathematics Institute of Hungian Academy of Sciences* 5 (1960), p. 1761.

[58] Paul Erdős and A. Rényi. "On random graphs". In: *Publicationes Mathematicae (Debrecen)* 6 (1959), p. 290. URL: /brokenurl#snap.stanford.edu/class/cs224w-readings/erdos60random.pdf.

[59] Jiaoe Wang et al. "Exploring the network structure and nodal centrality of China's air transport network: A complex network approach". In: *Journal of Transport Geography* 19.4 (2011), pp. 712–721. ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2010.08.012. URL: https://www.sciencedirect.com/science/article/pii/S0966692310001328.

[60] Stanley Wasserman, Katherine Faust, et al. "Social network analysis: Methods and applications". In: (1994).

[61] Danielle Bassett and Edward Bullmore. "Small-World Brain Networks Revisited". In: *The Neuroscientist* 23 (Aug. 2016). DOI: 10.1177/1073858416667720.

[62] Catharina Marie Stille et al. "Modeling the Mental Lexicon as Part of Long-Term and Working Memory and Simulating Lexical Access in a Naming Task Including Semantic and Phonological Cues". In: *Frontiers in Psychology* 11 (2020), p. 1594. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2020.01594. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2020.01594.

[63] Mark Steyvers and Joshua Tenenbaum. "The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth". In: *Cognitive science* 29 (Jan. 2005), pp. 41–78. DOI: 10.1207/s15516709cog2901_3.

[64] Haitao Liu and WenWen Li. "Language clusters based on linguistic complex networks". In: *Chinese Science Bulletin* 55 (Oct. 2010), pp. 3458–3465. DOI: 10.1007/s11434-010-4114-3.

[65] Andrew Goldberg and Xiaojin Zhu. "Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization". In: Jan. 2006.

[66] Monojit Choudhury and Animesh Mukherjee. "The Structure and Dynamics of Linguistic Networks". In: Feb. 2009, pp. 145–166. ISBN: 978-0-8176-4750-6. DOI: 10.1007/978-0-8176-4751-3_9.

[67] Ramon Ferrer i Cancho and Richard V. Solé. "The small world of human language". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268.1482 (2001), pp. 2261–2265. DOI: 10.1098/rspb.2001.1800. eprint: http://rspb.royalsocietypublishing.org/content/268/1482/2261.full.pdf+html. URL: http://rspb.royalsocietypublishing.org/content/268/1482/2261.abstract.

[68] R. Ferrer i Cancho, R. V. Solé, and R. Köhler. "Patterns in Syntactic Dependency Networks". In: *Physical Review E* 69 (2004), p. 051915.

[69] Adilson E. Motter et al. “Topology of the conceptual network of language”. In: *Physical Review E* 65.6 (June 2002). ISSN: 1095-3787. DOI: 10.1103/physreve.65.065102. URL: http://dx.doi.org/10.1103/PhysRevE.65.065102.

[70] Monojit Choudhury et al. “How Difficult is it to Develop a Perfect Spell-checker? A Cross-linguistic Analysis through Complex Network Approach”. In: (Apr. 2007).

[71] Mariano Sigman and Guillermo A. Cecchi. “Global organization of the Wordnet lexicon”. In: *Proceedings of the National Academy of Sciences* 99.3 (2002), pp. 1742–1747. ISSN: 0027-8424. DOI: 10.1073/pnas.022341799. eprint: https://www.pnas.org/content/99/3/1742.full.pdf. URL: https://www.pnas.org/content/99/3/1742.

[72] Barceló-Coblijn Lluís et al. “How Children Develop Their Ability to Combine Words : A Network-Based Approach”. In: *Adaptive Behavior* 27 (May 2019). DOI: 10.1177/1059712319847993.

[73] Yuyang Gao et al. “Comparison of directed and weighted co-occurrence networks of six languages”. In: *Physica A: Statistical Mechanics and its Applications* 393 (2014), pp. 579–589. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2013.08.075. URL: https://www.sciencedirect.com/science/article/pii/S037843711300825X.

[74] A. Masucci and Geoff Rodgers. “Differences between normal and shuffled texts: Structural properties of weighted networks”. In: *Advances in Complex Systems* 12 (Nov. 2011). DOI: 10.1142/S0219525909002039.

[75] Monojit Choudhury, Diptesh Chatterjee, and Animesh Mukherjee. “Global topology of word co-occurrence networks: Beyond the two-regime power-law”. In: vol. 2. Jan. 2010, pp. 162–170.

[76] William H. Fletcher. *Phrases in English - BNC N-Grams*. May 2010. URL: http://phrasesinenglish.org/explore.html.

[77] Quan Ha Le et al. “Extension of Zipf’s Law to Words and Phrases”. In: *COLING* (Mar. 2004). DOI: 10.3115/1072228.1072345.

[78] Seth Lloyd. “Measures of Complexity: A Nonexhaustive List”. In: *Control Systems Magazine, IEEE* 21 (Sept. 2001), pp. 7–8. DOI: 10.1109/MCS.2001.939938.

[79] Carlos Gershenson and Nelson Fernández. “Complexity and Information: Measuring Emergence, Self-organization, and Homeostasis at Multiple Scales”. In: *Complexity* 18 (Nov. 2012). DOI: 10.1002/cplx.21424.

[80] Peter Grassberger. “Toward a quantitative theory of self-generated complexity”. In: *International Journal of Theoretical Physics* 25 (Jan. 1986), pp. 907–938. DOI: 10.1007/BF00668821.

[81] J. Horgan. “From Complexity to Perplexity”. In: *Scientific American* (1995), pp. 74–79.

[82] M. Gell-Mann and S. Lloyd. “Effective Complexity”. In: SANTA FE INSTITUTE, 2003, pp. 387–398.

[83] Murray Gell-Mann and Seth Lloyd. “Information Measures, Effective Complexity, and Total Information”. In: *Complexity* 2.1 (1996), pp. 44–52.

[84] James W. McAllister. “Effective Complexity as a Measure of Information Content”. In: *Philosophy of Science* 70.2 (2003), pp. 302–307. DOI: 10.1086/375469.

[85] Ming Li and Paul M. B. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Texts and Monographs in Computer Science. Springer, 1993, pp. I–XX, 1–546. ISBN: 978-1-4757-3860-5.

[86] R.J. Solomonoff. "A formal theory of inductive inference. Part I". In: *Information and Control* 7.1 (1964), pp. 1–22. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(64)90223-2. URL: https://www.sciencedirect.com/science/article/pii/S0019995864902232.

[87] R.J. Solomonoff. "A formal theory of inductive inference. Part II". In: *Information and Control* 7.2 (1964), pp. 224–254. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(64)90131-7. URL: https://www.sciencedirect.com/science/article/pii/S0019995864901317.

[88] R. J. Solomonoff. "A preliminary report on a general theory of inductive inference". In: 1960.

[89] A. Kolmogorov. "Logical basis for information theory and probability theory". In: *IEEE Transactions on Information Theory* 14.5 (1968), pp. 662–664. DOI: 10.1109/TIT.1968.1054210.

[90] A. N. Kolmogorov. "Three approaches to the quantitative definition of information". In: *Problems of Information Transmission* 1.1 (1965), pp. 1–7.

[91] Gregory J. Chaitin. "On the Simplicity and Speed of Programs for Computing Infinite Sets of Natural Numbers". In: *J. ASSOC. COMPUT. MACH* 16 (1969), pp. 407–422.

[92] Gregory J. Chaitin. "On the Length of Programs for Computing Finite Binary Sequences". In: *Journal of the ACM* 13 (1966), pp. 547–569.

[93] Claude E. Shannon. "A mathematical theory of communication." In: *Bell Syst. Tech. J.* 27.3 (1948), pp. 379–423. URL: http://dblp.uni-trier.de/db/journals/bstj/bstj27.html#Shannon48.

[94] W. H. Zurek. "Algorithmic randomness and physical entropy". In: *Phys. Rev. A* 40 (8 Oct. 1989), pp. 4731–4751. DOI: 10.1103/PhysRevA.40.4731. URL: https://link.aps.org/doi/10.1103/PhysRevA.40.4731.

[95] Murray Gell-Mann. *The quark and the jaguar: adventures in the simple and the complex*. New York: W. H. Freeman, 1994. ISBN: 0716725819.

[96] A. C. Ross and G. Herdan. "Type-token mathematics : a textbook of mathematical linguistics". In: 1960.

[97] V. Balasubrahmanyan and S. Naranan. "Quantitative Linguistics and Complex System Studies." In: *Journal of Quantitative Linguistics* 3 (Dec. 1996), pp. 177–228. DOI: 10.1080/09296179608599629.

[98] V. Balasubrahmanyan and S. Naranan. "Algorithmic information, complexity and Zipf's law". In: *Glottometrics* 4 (2002), pp. 1–26.

[99] Chao-Lin Liu et al. *Character Distributions of Classical Chinese Literary Texts: Zipf's Law, Genres, and Epochs*. 2017. arXiv: 1709.05587 [cs.CL].

[100] Ke Jinyun. "Complex networks and human language". In: 2007.

[101] D.R. Amancio et al. “Using metrics from complex networks to evaluate machine translation”. In: *Physica A: Statistical Mechanics and its Applications* 390.1 (2011), pp. 131–142. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2010.08.052. URL: https://www.sciencedirect.com/science/article/pii/S0378437110007557.

[102] Jianyu Zheng et al. “Exploring Characteristics of Word Co-occurrence Network in Translated Chinese”. In: *2019 International Conference on Asian Language Processing (IALP)*. 2019, pp. 261–266. DOI: 10.1109/IALP48816.2019.9037722.

[103] Diego Raphael Amancio, Osvaldo N Oliveira Jr, and Luciano da Fontoura Costa. “Identification of literary movements using complex networks to represent texts”. In: *New Journal of Physics* 14.4 (2012), p. 043029.

[104] Christoforos Nalmpantis and Dimitris Vrakas. “Signal2Vec: Time Series Embedding Representation”. In: May 2019, pp. 80–90. ISBN: 978-3-642-54671-6. DOI: 10.1007/978-3-030-20257-6_7.

[105] Brad Ross and Prasanna Ramakrishnan. “song2vec: Determining Song Similarity using Deep Unsupervised Learning”. In: (2018).

[106] Xun Liang. “Social Computing Application in Unsupervised Oracle Handwriting Recognition Based on Pic2Vec Image Content Mapping”. In: *Social Computing with Artificial Intelligence*. Springer, 2020, pp. 249–255.

[107] Jaechoon Jo et al. “An Development of Image Retrieval Model based on Image2Vec using GAN”. In: *Journal of Digital Convergence* 16.12 (2018), pp. 301–307.

[108] Bill Bryson. *Uma Breve História de Quase Tudo*. Quetzal Editores, 2003.

[109] Bin Wang et al. “Evaluating word embedding models: methods and experimental results”. In: *APSIPA Transactions on Signal and Information Processing* 8 (2019). ISSN: 2048-7703. DOI: 10.1017/atsip.2019.12. URL: http://dx.doi.org/10.1017/ATSIP.2019.12.

[110] Tobias Schnabel et al. “Evaluation methods for unsupervised word embeddings”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 298–307. DOI: 10.18653/v1/D15-1036. URL: https://www.aclweb.org/anthology/D15-1036.

[111] José Camacho-Collados and Roberto Navigli. “Find the word that does not belong: A Framework for an Intrinsic Evaluation of Word Vector Representations”. In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 43–50. DOI: 10.18653/v1/W16-2508. URL: https://www.aclweb.org/anthology/W16-2508.

[112] Eneko Agirre et al. “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*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 19–27. URL: https://www.aclweb.org/anthology/N09-1003.

[113] Lukas Svoboda and Slobodan Beliga. *Evaluation of Croatian Word Embeddings*. 2017. arXiv: 1711.01804 [cs.CL].

Appendices

Appendix A

Dot Product Between Two Unit Vectors

A vector $\mathbf{v} = [v_1, v_2, ..., v_n]$ defined in a normed vector space is said to be a unit vector if $\|\mathbf{v}\| = 1$. The dot product between $\mathbf{v}$ and another vector $\mathbf{u} = [u_1, u_2, ..., u_n]$ is

$$\mathbf{v} \cdot \mathbf{u} = v_1 u_1 + v_2 u_2 + ... + v_n u_n \tag{A.1}$$

Let$E[\mathbf{v} \cdot \mathbf{u}]$ be the mean value of the dot product between two random unit vector in a real space of arbitrary dimension and let each individual component of $\mathbf{v}$ and $\mathbf{u}$ be an independent normal distributed random variable with mean 0 such that

$$E[v_1] = E[u_1] = E[v_2] = E[u_2] = ... = E[v_n] = E[u_n] = 0 \tag{A.2}$$

In that case it follows that

$$E[v_1 u_1] = E[v_2 u_2] = ... = E[v_n u_n] = 0 \tag{A.3}$$

And therefore,

$$E[v_1 u_1 + v_2 u_2 + ... + v_n u_n] = 0 \tag{A.4}$$

$$\implies E[\mathbf{v} \cdot \mathbf{u}] = 0 \tag{A.5}$$

The same conclusion is also achieved using a *brute force* method. With a simple python script, one computed 100 000 dot product between random unit vectors in a 200-dimensional real space which leads to the distribution in A.1.

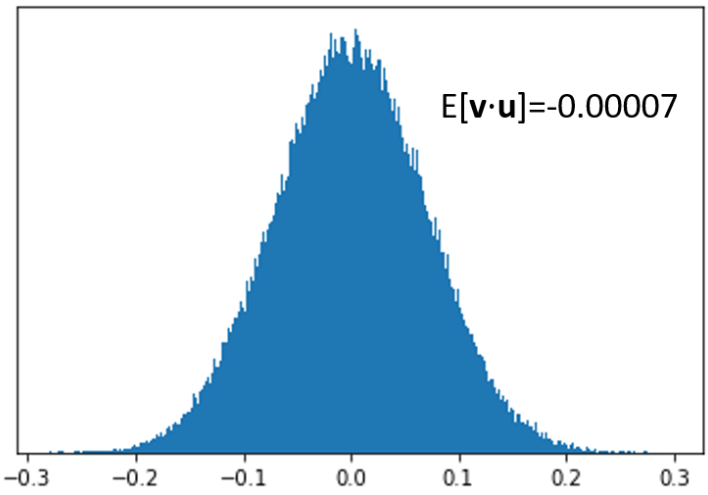

Figure A.1: Distribution of 100 000 dot product between random unit vectors in $\mathbb{R}^{200}$.

The experimental result of $E[\mathbf{v} \cdot \mathbf{u}] = -0.00007$ virtually means $E[\mathbf{v} \cdot \mathbf{u}] = 0$

Appendix B

Outlier Detection Source Code

```
#MAIN EVALUATION METHOD
def outlier_evaluation(Model, file):
    #file - path to  the outlier detection dataset file
    #word2vec model to be evaluated

    df = pd.read_csv(file, sep=';')
    correct=0.0
    for line in df.index:
        if( outlier(df["Word1"][line], df["Word2"][line], df["Word3"][line],
        df["Outlier"][line]), Model):
            correct=correct+1.0

    return correct/len(df.index) #evaluation score

#auxiliary method called by the main method
def cluster_sim(word1, word2, word3, Model):

    sim1=Model.wv.similarity(word1,word2)
    sim2=Model.wv.similarity(word1,word3)
    sim3=Model.wv.similarity(word2,word3)

    return sim1+sim2+sim3

#auxiliary method called by the main method
def outlier(word1, word2, word3, outlier, Model):

    cluster_scores = []
    cluster_scores.append(cluster_sim(word1, word2, outlier, Model))
    cluster_scores.append(cluster_sim(word1, outlier, word3, Model))
    cluster_scores.append(cluster_sim(outlier, word2, word3, Model))
    cluster_scores.append(cluster_sim(word1, word2, word3, Model))

    if np.argmax(cluster_scores)==3:
        return True
    else:
        return False
```

Appendix C

Model Validation

The quality of a word2vec model strongly depends on finding the set of hyperparameters that better fits the learning of the training corpus. Tuning the parameters can be essentially empirically testing combinations and check how the model scores in an evaluator. Yet, evaluating unsupervised learning is problematic because there is no ground truth to which the results can be meaningfully compared. More recently, some work has been developed on defining intrinsic and extrinsic word embedding evaluators [a] [109, 110].

It is not in the scope of this work to search for the perfect word2vec model, however, to validate and tune the model, two intrinsic evaluators were used:

Outlier Detection

This evaluator tests semantic coherence of vector space models by finding words that do not belong to a given group. Given a set of $W = \{w_1, w_2, ..., w_n\}$ that contains one outlier word and a compactness score $c(w)$ for words $w \in W$

$$c(w) = \sum_{w_i \in W \setminus w} \sum_{w_j \in W \setminus w, w_j \neq w_i} sim(w_i, w_j) \tag{C.1}$$

The outlier is the word with the highest compactness score [111].

The dataset used is composed by 189 four-word instances. Our implementation for the outlier detection evaluator can be found in appendix B.

Word Similarity

The goal of word similarity evaluation is to measure how well the notion of human perceived similarity is captured by the word vector representations. We used the *WordSim-353* similarity dataset contains 353 word pairs along with a human-assigned similarity score.

The evaluation metric is the Pearson's correlation between the human-assigned score and the word vector similarity $sim(w_1, w_2)$.

[a] Extrinsic evaluators use word embeddings as input features to a downstream task and measure changes in performance metrics specific to that task while intrinsic evaluators measure syntactic or semantic relationships among words directly comparing embeddings with human perception metrics.[109]

Outlier Detection Dataset			
Word 1	**Word** 2	**Word** 3	**Outlier**
brazil	colombia	uruguay	finance
microsoft	apple	intel	case
november	april	july	reported

Table C.1: Samples from the outlier detection dataset. The dataset was built from scratch for this work as no publicly available dataset was found. Full dataset on appendix E.

WordSim-353 Dataset		
Word 1	**Word** 2	**Score**
plane	car	5,77
sugar	approach	0.88
rock	jazz	7.59

Table C.2: Samples from the WordSim-353 similarity dataset [112].

Evaluation Results

The evaluation scores are shown in tables C.3 and C.4. Different window sizes, $c = \{5, 7, 9\}$, and dimensionalities, $N = \{100, 200, 300\}$, were tested for both Skip-Gram and CBOW architectures.

The scores reveal that despite skip-gram models scoring slightly over equivalent CBOW models, no combination of hyperparameters significantly outperforms or underperforms as the Outlier detection scores are consistently close to 100% and the Wordsim-353 scores are in line with other trained models in identical size corpus. [113]

Outlier Detection						
Architecture	**Skip-Gram**			**CBOW**		
Window Size	**5**	**7**	**9**	**5**	**7**	**9**
N=100	0.967	957	0.962	0.946	0.957	0.957
N=200	0.978	0.967	0.97	0.962	0.957	0.941
N=300	0.978	0.978	962	0.962	0.946	0.957

Table C.3: Outlier Detection evaluator scores.

WordSim-353						
Architecture	**Skip-Gram**			**CBOW**		
Window Size	**5**	**7**	**9**	**5**	**7**	**9**
N=100	0.451	0.457	0.475	0.406	0.408	0.4130
N=200	0.466	0.459	0.474	0.404	0.406	0.421
N=300	0.478	0.480	0.482	0.416	0.393	0.439

Table C.4: Word similarity evaluator scores.

Appendix D

Randomize Methods

```
import random

def shuffle(input_list, count=2):
    '''Shuffles any n number of values in a list'''
    indices_to_shuffle = random.sample(range(len(input_list)), k=count)
    to_shuffle = [input_list[i] for i in indices_to_shuffle]
    random.shuffle(to_shuffle)
    for index, value in enumerate(to_shuffle):
        old_index = indices_to_shuffle[index]
        input_list[old_index] = value
    return input_list

def total_randomize_sentences(sentences):
    sizes=[]
    words=[]
    random_sentences=[]

    for sentence in sentences:
        sizes.append(len(sentence))

    for sentence in sentences:
        for word in sentence:
            words.append(word)

    random.shuffle(words)

    counter=0

    for size in sizes:
        random_sentence=[]
        for i in range(size):
            random_sentence.append(words[counter])
            counter+=1
        random_sentences.append(random_sentence)

    return random_sentences
```

```
37
38  def randomize_sentences(sentences, random_samples):
39      sizes=[]
40      words=[]
41      random_sentences=[]
42
43      for sentence in sentences:
44          sizes.append(len(sentence))
45
46      for sentence in sentences:
47          for word in sentence:
48              words.append(word)
49
50      shuffle(words,random_samples)
51
52      counter=0
53
54      for size in sizes:
55          random_sentence=[]
56          for i in range(size):
57              random_sentence.append(words[counter])
58              counter+=1
59          random_sentences.append(random_sentence)
60
61      return random_sentences
```

Appendix E

Outlier Detection Set

	Word1	Word2	Word3	Outlier Word
1	cat	dog	mouse	flower
2	pig	sheep	cow	foot
3	eagle	falcon	hawk	pink
4	snake	anaconda	python	funny
5	bird	dog	mouse	europe
6	goat	horse	cow	onion
7	blue	red	green	big
8	purple	yellow	brown	wine
9	brown	red	pink	father
10	brown	purple	green	ensemble
11	purple	yellow	orange	masterpiece
12	yellow	red	pink	sequel
13	dollar	euro	yen	violin
14	dollar	pound	yen	battlefield
15	comedy	crime	drama	guns
16	action	adventure	comedy	iranians
17	drama	horror	thriller	court
18	comedy	romance	drama	another
19	action	adventure	western	india
20	western	romance	musical	firm
21	mother	father	brother	earlier
22	grandmother	father	grandfather	text
23	vodka	whisky	gin	support
24	fries	burger	pizza	going
25	pineapple	grape	orange	help
26	wine	beer	cider	statement
27	beef	lamb	pork	lower
28	tomato	potato	onion	my

	Word1	Word2	Word3	Outlier Word
29	soda	burger	pizza	good
30	pineapple	mango	banana	major
31	tomato	garlic	onion	need
32	spain	italy	portugal	likely
33	brazil	colombia	argentina	fund
34	bolivia	chile	venezuela	former
35	madrid	barcelona	bilbao	plans
36	sweden	finland	denmark	city
37	croatia	belgium	france	index
38	california	montana	nebraska	right
39	massachusetts	wisconsin	florida	however
40	washington	michigan	utah	meeting
41	china	japan	korea	here
42	egypt	syria	iran	security
43	kenya	ethiopia	ghana	increase
44	johannesburg	pretoria	durban	trading
45	manchester	liverpool	london	number
46	brighton	newcastle	southampton	energy
47	africa	europe	asia	big
48	america	asia	africa	credit
49	germany	italy	portugal	due
50	brazil	colombia	uruguay	finance
51	uruguay	chile	paraguay	power
52	mallorca	barcelona	bilbao	management
53	sweden	iceland	denmark	among
54	germany	belgium	slovakia	think
55	california	massachusetts	washington	election
56	montana	wisconsin	michigan	early
57	nebrasca	michigan	alabama	called
58	china	Pakistan	india	go
59	afghanistan	iraq	iran	used
60	kenya	ethiopia	ghana	less
61	rome	paris	berlin	better
62	greek	italian	french	cut
63	american	european	asian	health
64	africa	oceania	asia	run
65	america	asia	oceania	rose
66	pacific	atlantic	indian	move
67	guitar	bass	drums	cash
68	piano	drums	bass	revenue
69	microsoft	apple	intel	case

	Word1	Word2	Word3	Outlier Word
70	bmw	mercedes	audi	future
71	nissan	mazda	toyota	following
72	renault	peugeot	citroen	start
73	porsche	opel	bmw	plan
74	ferrari	lamborghini	maserati	whether
75	samsung	lg	huawei	hit
76	xiaomi	huawei	lenovo	largest
77	adidas	nike	puma	sector
78	youtube	google	amazon	days
79	facebook	twitter	youtube	already
80	instagram	snapchat	whatsapp	today
81	linkedin	facebook	twitter	far
82	microsoft	IBM	intel	too
83	bmw	volkswagen	audi	countries
84	nissan	honda	toyota	private
85	porsche	peugeot	citroen	current
86	renault	opel	volkswagen	political
87	ferrari	mclaren	fiat	open
88	samsung	xiamo	huawei	cost
89	xiaomi	huawei	nokia	demand
90	facebook	twitter	linkedin	director
91	instagram	vine	pinterest	old
92	youtube	reddit	twitter	free
93	gucci	chanel	vuitton	biggest
94	mile	foot	yard	microsoft
95	astrazeneca	pfizer	merck	making
96	gucci	dior	sephora	britain
97	gold	silver	copper	cent
98	zinc	steel	iron	decision
99	platinum	aluminium	zinc	want
100	bronze	silver	copper	points
101	zinc	copper	iron	inflation
102	platinum	aluminium	titanium	fed
103	one	two	three	income
104	ten	eleven	twelve	europe
105	twenty	thirty	forty	agreement
106	three	four	five	times
107	first	second	third	service
108	twentieth	nineteenth	eighteenth	ltd
109	ninety	eighty	sixty	ratings

	Word1	Word2	Word3	Outlier Word
110	south	west	north	costs
111	east	west	south	best
112	lunch	dinner	breakfast	office
113	big	small	medium	point
114	big	huge	large	each
115	liver	heart	lung	looking
116	plane	jet	helicopter	september
117	teacher	doctor	lawyer	total
118	rose	sunflower	daisy	across
119	happiness	sadness	joy	value
120	pants	jeans	trousers	system
121	shirt	sweater	jacket	half
122	king	queen	prince	reported
123	princess	prince	queen	additional
124	harvard	yale	stanford	qatar
125	lieutenant	captain	officer	rs
126	hydrogen	helium	lithium	research
127	helium	argon	neon	least
128	south	east	north	media
129	east	west	north	official
130	morning	afternoon	evening	products
131	large	tall	medium	equity
132	small	tiny	large	know
133	liberalism	communism	socialism	line
134	liver	heart	stomach	white
135	physics	astrology	mathematics	put
136	biology	chemistry	physics	customers
137	kitchen	bedroom	bathroom	continue
138	federer	nadal	raonic	quotes
139	federer	thiem	djokovic	minutes
140	obama	bush	clinton	net
141	merkel	obama	hollande	corp
142	vettel	senna	hamilton	offer
143	hamilton	raikkonen	bottas	available
144	dicaprio	pitt	damon	securities
145	musk	jobs	gates	rise
146	federer	wawrinka	murray	production
147	wawrinka	thiem	djokovic	reports
148	bush	clinton	Reagen	small
149	obama	bush	clinton	might
150	merkel	rajoy	hollande	level

	Word1	Word2	Word3	Outlier Word
151	vettel	ricciardo	hamilton	rating
152	vettel	raikkonen	verstappen	found
153	zuckerberg	bezos	musk	announced
154	zuckerberg	jobs	gates	analysts
155	mars	mercury	venus	full
156	jupiter	saturn	neptune	senior
157	earth	mercury	venus	short
158	jupiter	saturn	mercury	given
159	football	baseball	basketball	union
160	hockey	tennis	golf	few
161	boxing	judo	triathlon	become
162	football	tennis	judo	write
163	voleyball	baseball	basketball	later
164	soccer	tennis	golf	large
165	boxing	football	triathlon	little
166	chess	tennis	judo	street
167	monday	tuesday	wednesday	nine
168	saturday	sunday	friday	command
169	thursday	friday	monday	qatar
170	january	march	april	research
171	february	december	november	least
172	april	june	may	media
173	november	april	july	euro
174	april	october	february	across
175	day	week	month	several
176	week	month	year	bill
177	monday	tuesday	friday	left
178	monday	sunday	friday	buy
179	thursday	sunday	monday	results
180	january	march	august	north
181	february	august	november	department
182	december	june	may	average
183	november	april	july	reported
184	january	october	november	seen
185	century	decade	year	funds
186	yesterday	today	tomorrow	october
187	meter	centimeter	kilometer	baseball
188	celsius	fahrenheit	kelvin	alphabet