NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENTO DE FÍSICA

LAURA RITA MARTINS ELVAS

Licenciatura em Ciências da Engenharia

# ESTUDOS DE VARIAÇÃO DE DENSIDADE NUM ESPAÇO INFLACIONÁRIO

MESTRADO EM ENGENHARIA FÍSICA

Universidade NOVA de Lisboa
Novembro, 2021

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENTO DE FÍSICA

# ESTUDOS DE VARIAÇÃO DE DENSIDADE NUM ESPAÇO INFLACIONÁRIO

**LAURA RITA MARTINS ELVAS**

Licenciatura em Ciências da Engenharia

**Coorientadores:** João Pires da Cruz
*Professor Auxiliar Convidado, FCUL*

André Wemans
*Professor Auxiliar, NOVA/SST*

**Júri**

**Presidente:** Doutora Maria de Fátima Guerreiro da Silva Campo Raposo
*Professora Associada com Agregação, NOVA/SST*

**Arguente:** Doutor Hygor Piaget de Melo
*Investigador pós-doutorado, CFTC Ciências Universidade de Lisboa*

**Vogal:** Doutor João Pires da Cruz
*Professor Auxiliar Convidado, FCUL*

MESTRADO EM ENGENHARIA FÍSICA

Universidade NOVA de Lisboa
Novembro, 2021

**Estudos de Variação de Densidade num Espaço Inflacionário**



Este documento foi gerado com o processador (pdf/Xe/Lua)LaTeX e o modelo NOVAthesis (v6.9.5) [1].

# Agradecimentos

A finalização de uma dissertação marca um momento importante na vida de qualquer estudante universitário. É, talvez, o maior trabalho e o de maior peso que se faz ao longo do curso. É, definitivamente, o mais difícil e aquele em que nos dedicamos com mais alma e coração. É um esforço coletivo, apesar de na capa surgir apenas um autor.

Em primeiro lugar, queria agradecer aos meus co-orientadores, João Cruz e professor André Wemans, por toda a ajuda que me forneceram ao longo deste trabalho. Ao João, em especial, pelo conselho que me deu numa das nossas reuniões. Não o esquecerei.

Queria, de seguida, agradecer à faculdade NOVA/SST por todo o conhecimento e apoio oferecido ao longo do curso.

Por último, queria agradecer a três pessoas, todas com um lugar especial no meu coração: os meus pais e o meu namorado. Aos meus pais, Vanda e Mário, por todos os sacrifícios que fizeram para poder chegar onde cheguei. Por todas as séries que não vimos, por todos os locais que não fomos e por todo o stress que passámos por causa do estudo e do curso. Ao meu namorado, César, por me ter mantido sã durante todo este processo. Só as tuas palavras eram suficientes para me acalmarem em momentos de pânico e me ajudarem a superar os obstáculos. Por isto e por muito mais, obrigada a todos.

*«I have no special talent. I am only passionately curious.» (Albert Einstein)*

# Resumo

O estudo dos sistemas em expansão é, de momento, uma área onde se estão a colocar grandes esforços. Dentro dos sistemas desta natureza, o mais simples de estudar é o espaço de palavras devido à sua acessibilidade. A contrapartida do espaço de palavras é este não ser um espaço plano e haver a necessidade de o transformar num que possa ser estudado.

O trabalho aqui descrito utilizou um modelo *Word2Vec* para fazer a projeção do espaço num espaço de Hilbert e obter a matriz densidade associada à projeção do espaço das palavras sobre aquele. A partir da matriz, estuda-se a densidade para vários períodos de tempo e conjuntos de texto.

Os resultados obtidos demonstraram uma descida da densidade seguida de uma subida com oscilações. Em alguns casos, a densidade sofreu uma descida até atingir um limite onde estabilizou. Os dados registados foram surpreendentes na medida em que o esperado era um crescimento na densidade com o aumento de palavras no espaço o que comprova que ainda não se conhece totalmente o comportamento do mesmo e que mais estudos dentro do assunto são necessários.

**Palavras-chave:** Espaço de palavras, Sistema em expansão, Densidade, *Machine learning*.

# Abstract

The study of expanding systems is, at the moment, an area where great efforts are being put. Within systems of this nature, the simplest to study is the word space due to its accessibility. The problem with the word space is that it is not a flat space and that gives rise to the need of transforming it into one that can be studied.

The work described here used a Word2Vec model to project the space onto a Hilbert space and obtain the density matrix associated with the projection of the word space over the resulting Hilbert space. From the matrix, the density is studied for various periods and text sets.

The results obtained showed a decrease in density followed by a rise with oscillations. In some cases, the density declined until it reached a limit where it stabilized. The data recorded was surprising since it was expected an increase of density with the increase of words in the space, which proves that the space behavior is not yet fully known and that more studies on the subject are needed.

**Keywords:** Word space, Expanding system, Density, Machine learning.

# Índice

# Índice de Figuras

# Índice de Tabelas

# Índice de Listagens

# Siglas

**ML** *Machine Learning*

**OHE** *One-Hot Encoding*

**PLN** Processamento de Linguagem Natural

**RNAs** Redes Neuronais Artificiais

**SG** *Skip-gram*

**SNE** *Stochastic Neighbor Embedding*

**t-SNE** *t-Distributed Stochastic Neighbor Embedding*

Capítulo 1

# Introdução

Segundo a definição do *Dicionário Priberam da Língua Portuguesa* [2], a definição mais convencional de densidade consiste na "relação entre a massa ou o peso de um corpo com o seu volume (...)". Considerando esta definição, a densidade, sendo a razão entre duas propriedades extensivas, é, por sua vez, uma propriedade intensiva. Uma outra definição e mais geral resume-se à razão entre a quantidade de algo e um volume [3].

A medição da densidade é aplicada, maioritariamente, na identificação de substâncias puras ou a composição de misturas. Regra geral, as densidades mais baixas pertencem aos gases, enquanto que as densidades mais altas correspondem aos sólidos, apesar destes terem uma variação elevada. Na zona intermédia encontram-se os líquidos. Porém, há sempre exceções. Dentro da indústria dos refrigerantes, a medição da densidade dos mesmos permite determinar a concentração de açúcar presente na bebida. Já na indústria petroquímica a densidade assiste na determinação da qualidade dos produtos produzidos, assim como na indústria das bebidas alcoólicas [4]. Um exemplo mais trivial da utilização da densidade, nomeadamente, da variação de densidade com a temperatura são as lâmpadas de lava.

Colocando de lado a densidade de substâncias, há outros casos onde se pode medir a densidade de algo mais abstrato. Por exemplo, nos semicondutores é interessante saber a sua densidade de estados para determinar o hiato entre a banda de condução e a banda de valência. A densidade de estados resume-se à quantidade de estados possíveis de serem ocupados pelos eletrões num determinado nível energético, ou seja, é o número de estados de eletrões por unidade de volume por unidade de energia. Esta densidade está intimamente relacionada com o calor específico e a suscetibilidade magnética [5].

Contudo, com o crescimento da utilização de *Machine Learning* (ML) no mundo, novos espaços de estudo vão surgindo e, com eles, novos aspectos para examinar. Estes modelos possuem a vantagem de conseguirem analisar quantidade de dados muito superiores às dos modelos tradicionais acompanhados pelo facto de produzirem previsões mais confiáveis e precisas [6]. Uma das áreas que beneficia destes modelos é a da medicina. Com a atual pandemia global, estão a ser desenvolvidos estudos que utilizam esta tecnologia para criar modelos predicativos de casos severos de COVID-19 em doentes [7] e Tiwari et al. [8] tenta desenvolver um novo índice de vulnerabilidade para os Estados Unidos da América.

Dentro do âmbito económico, são utilizados algoritmos de ML para analisar o contágio financeiro dentro de uma rede financeira criada através de correlações [6]. Um outro trabalho relacionado com o tema é o de Aromi [9] que analisa textos de teor financeiro com o objetivo de construir indicadores de incerteza.

Considerando a análise de texto por ML, os resultados gerados pelos algoritmos, normalmente associados a redes neuronais, são mapas de palavras aglomeradas tendo em conta a sua semelhança tanto sintáctica como semântica. Quanto maiores são as amostras de texto, maior é o número de palavras, o número de relações que se estabelecem e o tamanho do espaço. As relações que se estabelecem estão, não só dependentes das duas palavras em causa, como também das palavras presentes dentro do espaço.

O interesse de estudar os mapas de palavras e o seu espaço de um ponto de vista físico é a sua semelhança com os sistemas em expansão. A forma como o espaço é estudado é considerando uma separação dos dados em partes, quer através dos períodos a que correspondem os dados quer através do número de palavras que existem neles, e indo adicionando as várias partes a um modelo *Word2Vec* já criado e treinado com as partes anteriores. A constante adição de novas palavras ao modelo cria, efetivamente, um sistema em expansão cuja vantagem é ser composto por objetos mais intuitivos e acessíveis de estudar em oposição, por exemplo, ao universo físico em inflação. Com isto em mente, o objectivo deste trabalho é estudar e compreender o comportamento da densidade dentro do espaço de palavras e, consequentemente, dentro dos espaços inflacionários.

Capítulo 2

# Conceitos Teóricos

Neste capítulo é descrita toda a teoria mencionada ao longo do projeto, encontrando-se separado em duas secções. A primeira explica os conceitos base necessários à boa compreensão do que se realizou no trabalho enquanto que a última explicara o espaço das palavras e faz ligação entre o mesmo e os conceitos base.

## 2.1 Conceitos Base

Os conceitos aqui apresentados são a base e o mínimo necessário ao entendimento do projeto que se realizou. O primeiro conceito referido é o de espaço de Hilbert que é o espaço vetorial associado aos *word embeddings* ou vetores representativos de palavras num vocabulário. O segundo conceito é o da matriz densidade que é uma forma alternativa de representar um estado. O terceiro conceito é sobre as Redes Neuronais Artificiais (RNAs) e o quarto é uma explicação em detalhe do modelo utilizado no projeto. Entender todos estes conceitos facilitará o entendimento da secção final. É de notar que neste trabalho a utilização de formalismos associados à mecânica quântica não é uma demonstração das palavras serem objetos quânticos. A utilização de parte do racional é para um melhor entendimento dos fenómenos.

### 2.1.1 Espaços de Hilbert

Dentro da mecânica quântica, um conceito comummente referido é o de espaço de Hilbert. A sua importância deriva do facto de qualquer estado de um sistema quântico poder ser aqui representado por uma função ou funcional representada como um vector naquele espaço. Estes espaços, cujo nome foi dado em honra de David Hilbert, podem ser tanto de dimensões finitas como infinitas, porém e no âmbito do projeto que se desenvolveu, o foco é no tratamento do espaço com uma dimensão finita. A notação utilizada é a notação

de Dirac [10].

Em primeiro lugar, um espaço de Hilbert, $\mathcal{H}$, é um espaço vetorial sobre $\mathbb{R}$ ou $\mathbb{C}$ e, como tal, para qualquer vetor $|\psi\rangle, |\phi\rangle, |\theta\rangle \in \mathcal{H}$ são obedecidas as propriedades comutativas, associativas e a do elemento neutro da adição (eqs. 2.1) [10, 11].

$$\begin{aligned} |\psi\rangle + |\phi\rangle &= |\phi\rangle + |\psi\rangle \\ (|\psi\rangle + |\phi\rangle) + |\theta\rangle &= |\psi\rangle + (|\phi\rangle + |\theta\rangle) \\ |\psi\rangle + |0\rangle &= |\psi\rangle \end{aligned} \tag{2.1}$$

Para além disso, para cada vetor $|\psi\rangle$, existe um vetor inverso, $|-\psi\rangle \in \mathcal{H}$, para o qual se verifica a condição expressa pela eq. 2.2.

$$|\psi\rangle + |-\psi\rangle = |0\rangle \tag{2.2}$$

No caso da multiplicação, para qualquer $a, b \in \mathbb{K}$, com $\mathbb{K} = \mathbb{R}, \mathbb{C}$ é obedecida a propriedade distributiva, tanto no domínio $\mathcal{H}$ como em $\mathbb{K}$, a propriedade associativa e a propriedade do elemento neutro (eqs. 2.3).

$$\begin{aligned} a(|\psi\rangle + |\phi\rangle) &= a|\phi\rangle + a|\psi\rangle \\ (a+b)|\psi\rangle &= a|\psi\rangle + b|\psi\rangle \\ ab|\psi\rangle &= a(b|\psi\rangle) \\ 1|\psi\rangle &= |\psi\rangle \end{aligned} \tag{2.3}$$

Em adição, o produto interno encontra-se definido dentro dos espaços de Hilbert, sendo denotado por $\langle .|. \rangle$. A aplicação $\langle .|. \rangle : \mathcal{H} \times \mathcal{H} \longmapsto \mathbb{K}, \mathbb{K} = \mathbb{R}, \mathbb{C}$, beneficia de uma simetria, conjugada ou não consoante o domínio, assim como de linearidade, total ou só com respeito ao *ket*, aditividade e positividade (eqs. 2.4 e 2.5).

$$\begin{aligned} \langle .|. \rangle : \mathcal{H} \times \mathcal{H} &\longmapsto \mathbb{R} \\ \langle \psi \mid \phi \rangle &= \langle \phi \mid \psi \rangle \\ \langle \psi \mid a\phi \rangle &= \langle a\psi \mid \phi \rangle = a\langle \psi \mid \phi \rangle \end{aligned} \qquad \begin{aligned} \langle .|. \rangle : \mathcal{H} \times \mathcal{H} &\longmapsto \mathbb{C} \\ \langle \psi \mid \phi \rangle &= \overline{\langle \phi \mid \psi \rangle} \\ \langle \psi \mid a\phi \rangle &= a\langle \psi \mid \phi \rangle \end{aligned} \tag{2.4}$$

$$\begin{aligned} \langle .|. \rangle : \mathcal{H} \times \mathcal{H} &\longmapsto \mathbb{K}, \mathbb{K} = \mathbb{R}, \mathbb{C} \\ \langle \psi \mid \phi + \theta \rangle &= \langle \psi \mid \phi \rangle + \langle \psi \mid \theta \rangle \\ \langle \psi \mid \psi \rangle &\geq 0 \quad (\langle \psi \mid \psi \rangle = 0 \Rightarrow |\psi\rangle = 0) \end{aligned} \tag{2.5}$$

A definição do produto interno dentro dos espaços de Hilbert e as propriedades expressas nas eqs. 2.4 e 2.5 acarretam várias consequências:

- No caso em que $\mathbb{K} = \mathbb{C}$, o produto interno é antilinear com respeito ao *bra*, eq. 2.6.

$$\langle a\psi \mid \phi \rangle = \overline{a}\langle \psi \mid \phi \rangle \tag{2.6}$$

- $\langle \psi \mid \psi \rangle = \overline{\langle \psi \mid \psi \rangle}$ é necessariamente um número real.
- A norma de um vetor encontra-se definida, eq. 2.7.

$$\| |\psi\rangle \| = \sqrt{\langle \psi \mid \psi \rangle} \tag{2.7}$$

- Garante-se a desigualdade de Cauchy–Schwarz, eq 2.8.

$$|\langle \psi \mid \psi \rangle|^2 \leq \langle \psi \mid \psi \rangle \langle \phi \mid \phi \rangle \tag{2.8}$$

No caso de um espaço de Hilbert de dimensão finita, a sua dimensão é igual ao maior número de vetores linearmente independentes que existe dentro desse espaço. Um conjunto de vetores $\{|\psi_1\rangle, |\psi_2\rangle, ..., |\psi_n\rangle\}$ é linearmente independente se a única solução para a eq. 2.9 for $c_1, c_2, ..., c_n = 0$.

$$c_1 |\psi_1\rangle + c_2 |\psi_2\rangle + ... + c_n |\psi_n\rangle = 0 \tag{2.9}$$

O mesmo conjunto de vetores $\{|\psi_1\rangle, |\psi_2\rangle, ..., |\psi_n\rangle\}$ é ainda ortonormal se satisfazer a condição 2.10, tornando-se uma base do espaço $\mathcal{H}$ se qualquer $|\phi\rangle \in \mathcal{H}$ puder ser descrito como uma combinação única $|\phi\rangle = \sum_{a=1}^{n} c_a |\psi_a\rangle$, com $c_a \in \mathbb{R}, \mathbb{C}$.

$$\langle \psi_i \mid \psi_j \rangle = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2.10}$$

Para o caso em que não exista um número finito de vetores linearmente independentes, o espaço de Hilbert passa a ser um espaço vetorial de dimensão infinita e torna-se necessário exigir que o espaço seja completo, isto é, que todas as sequências de Cauchy sejam convergentes em $\mathcal{H}$ [10, 11].

### 2.1.2 Matriz Densidade

Ao longo da subsecção 2.1.1 foi sempre utilizada a notação de Dirac em *bras* e *kets* como forma de representar vetores e, consequentemente, estados. Esta representação é, geralmente, a primeira a ser introduzida quando ainda se está a aprender mecânica quântica devido à sua simplicidade. É especialmente útil quando o sistema a estudar pode ser descrito por um estado quântico ou por uma sobreposição de estados quânticos. Contudo, quando o sistema é uma mistura de estados esta notação deixa de ser utilizada, pois torna-se difícil usar. Nestes casos, o ideal é usar a forma matricial ou o operador densidade $\hat{\rho}$ para descrever estatisticamente o sistema [12].

Para estados mistos, $\hat{\rho}$ é descrito pela eq. 2.11. O termo $p_i$ é a probabilidade de se escolher o estado $|\phi_i\rangle$, enquanto que o termo $|\phi_i\rangle\langle\phi_i|$ é o operador projeção que projeta um estado no estado $|\phi_i\rangle$, devolvendo um estado proporcional ao estado de referência. Contudo, para os estados puros como $|\phi_i\rangle$ o operador densidade é igual ao operador de projeção, logo $|\phi_i\rangle\langle\phi_i|$ pode ser substituído por $\hat{\rho}_i$, o operador densidade associado a cada estado presente no sistema. Resumidamente, o operador densidade é a média dos operadores densidade dos estados puros.

$$\hat{\rho} = \sum_i p_i |\phi_i\rangle\langle\phi_i| = \sum_i p_i \hat{\rho}_i \tag{2.11}$$

As propriedades do operador alteram-se consoante o operador é utilizado em estados puros ou em estados mistos. Para estados puros, o operador é hermítico e idempotente, eq. 2.12.

$$\begin{aligned} \text{Hermítico} \quad & \hat{\rho} = \hat{\rho}^\dagger \\ \text{Idempotente} \quad & \hat{\rho}^2 = \hat{\rho} \end{aligned} \tag{2.12}$$

A forma como o operador é definido permite que não haja ambiguidade associada à fase do estado, isto é, se a um estado $|\phi\rangle$ está associado o operador $\hat{\rho}_\phi$ e ao estado $|\psi\rangle$ está associado o operador $\hat{\rho}_\psi$ e $|\psi\rangle = e^{i\theta}|\phi\rangle$, os operadores $\hat{\rho}_\phi$ e $\hat{\rho}_\psi$ são iguais. Para os estados mistos, o operador $\hat{\rho}$ deixa de ser idempotente, logo deixa de ser um operador de projeção.

A partir do operador densidade, operações como os cálculos de valores esperados, evolução temporal e normalização passam a ser realizados considerando o traço do operador, $Tr(\hat{\rho})$, eq. 2.13, o que é muito mais simples do que a utilização de *kets* e *bras*.

$$\begin{aligned} \text{Normalização} \quad & Tr(\hat{\rho}) = 1 \\ \text{Valor Esperado} \quad & \langle\hat{A}\rangle = Tr(\hat{\rho}\hat{A}) \\ \text{Evolução Temporal} \quad & \frac{d}{dt}\hat{\rho}(t) = \frac{1}{i\hbar}[\hat{H}(t), \hat{\rho}(t)] \end{aligned} \tag{2.13}$$

A distinção entre os dois operadores densidade, o para os estados puros e para os estados mistos pode ser feita considerando não só o facto de $\hat{\rho}$ ser apenas idempotente para os estados puros mas também pelo facto de $Tr(\hat{\rho}^2) = 1$ para os estados puros e $Tr(\hat{\rho}^2) < 1$ para os estados mistos.

### 2.1.3 Redes Neuronais Artificiais

O Processamento de Linguagem Natural (PLN) é um ramo da inteligência artificial que analisa e tenta compreender a linguagem natural com o objetivo de tornar a comunicação homem-máquina mais intuitiva para o ser humano. As maiores dificuldades dentro desta área passam por ensinar o computador a identificar expressões idiomáticas, metáforas,

sarcasmo, variações estruturais dentro de frases, etc [13]. Para tentar ultrapassar estas dificuldades, são habitualmente desenvolvidos algoritmos baseados nas RNAs cuja característica mor é a sua semelhança com o cérebro humano.

Tal como o cérebro humano, as RNAs são constituídas por unidades computacionais simples, também designadas por neurónios, que comunicam entre si através do envio de sinais. Estas unidades podem ser separadas nas seguintes categorias [14]:

- Unidades de *input*: são as unidades encarregadas de receber a informação proveniente do exterior e compõem a primeira camada da rede.
- Unidades ocultas: incorporam a camada oculta da rede. O seu *input* é o *output* ponderado das unidades da camada anterior mais um *offset*. O valor obtido é enviado para uma função de ativação que determina se a unidade deve ser ativada ou não. As unidades ativas enviam o seu *output* para a camada seguinte.
- Unidade de *output*: pertencem à camada de *output*, enviando o seu sinal para fora da rede.

A figura 2.1 ilustra uma unidade onde o cálculo do *input* efetivo, ou regra de propagação, é através de uma soma ponderada dos *inputs* das unidades anteriores.

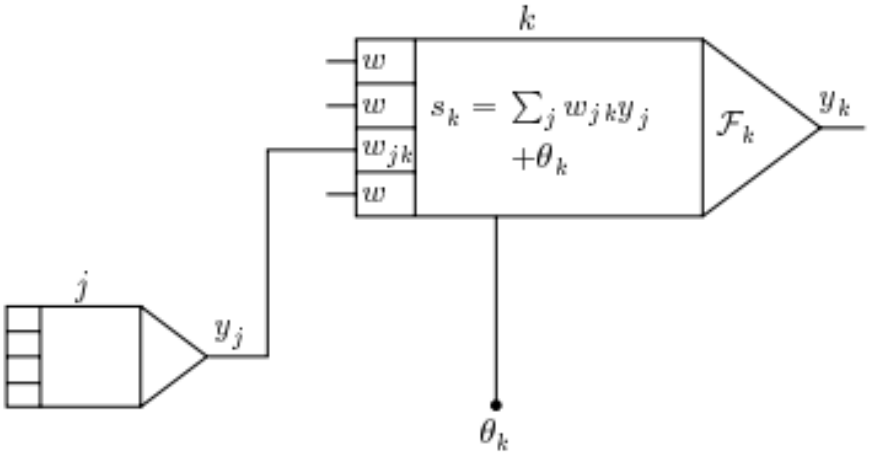


Figura 2.1: Representação dos componentes básicos de uma rede neuronal onde a regra de propagação usada é uma soma ponderada. A unidade $k$ recebe de cada unidade $j$ o seu *input* $y_j$ e faz uma soma ponderada $s_k$ tendo em conta os pesos $w_{ij}$ e o seu *offset* $\theta_k$. O valor obtido é utilizado na função de ativação $F_k$, o seu estado de ativação é determinado e o *output* $y_k$ devolvido. Retirada de [14].

Devido à sua estrutura e funcionamento, as RNAs utilizam uma representação distribuída para representar conceitos. Cada conceito está associado a um conjunto de unidades, formando um padrão de ativação. Por sua vez, as unidades que constituem o padrão não estão restringidas a este e podem participar na formação de outros padrões, fazendo

parte da representação de outros conceitos. No caso das palavras e da sua vetorização, a representação distribuída é aplicada à forma como os seus vetores são codificados.

Não considerando a representação distribuída, a técnica mais simples de aplicar na transformação de palavras em vetores é a *One-Hot Encoding* (OHE). Esta técnica de representação local cria vetores cuja dimensão é igual ao número de palavras do vocabulário a representar, sendo todas as suas entradas 0 à exceção de uma que é 1. A posição da entrada com valor um é diferente para cada palavra. A título de exemplo, considere-se a seguinte citação de Albert Einstein:

"The distinction between the past, present and future is only a stubbornly persistent illusion."

Tabela 2.1: Vetorização das palavras dentro da citação de Einstein utilizando a técnica *One-Hot Encoding*.

| **Posição** | **The** | **distinction** | **between** | **past** | **present** | **...** | **illusion** |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | ... | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | ... | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | ... | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | ... | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | ... | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 12 | 0 | 0 | 0 | 0 | 0 | ... | 1 |

Na tabela 2.1 estão representadas algumas das palavras presentes na citação e os vetores a elas associados. Cada um destes vetores tem uma dimensão igual ao número de palavras distintas na frase, que são treze, e apenas uma entrada não neutra cuja posição vai avançando à medida que a palavra a representar se encontra mais para o final da frase. É de notar que a palavra *the*, apesar de surgir duas vezes na frase, não possui duas entradas com o valor 1. Os valores 1 e 0 servem apenas para identificar as diferentes palavras dentro da frase e não para contabilizar as vezes que surgem dentro da mesma.

Analisando melhor a tabela 2.1, a abordagem OHE apresenta pelo menos dois problemas evidentes:

1. A representação de um vocabulário com $n$ palavras requer vetores de $n$ dimensões. Os vocabulários geralmente utilizados em PLN têm entre milhares e milhões de palavras, obrigando a que os vetores correspondentes tenham também dimensões igualmente elevadas, figura 2.2.

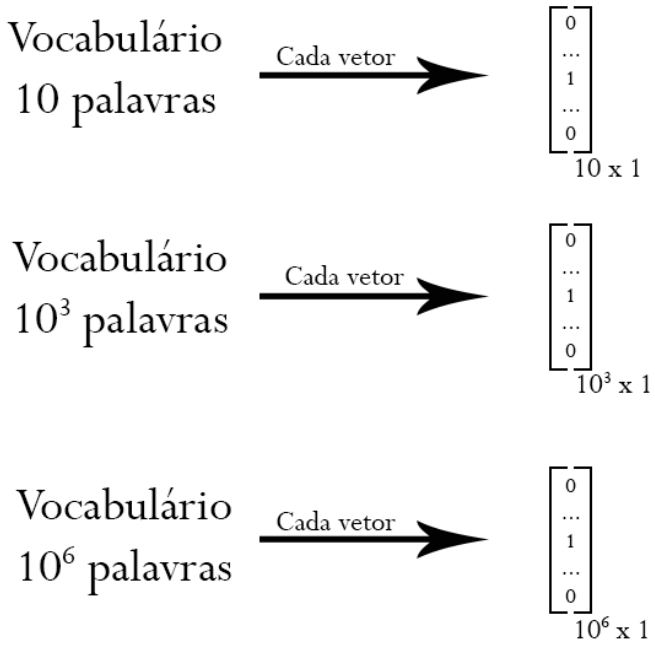


Figura 2.2: Variação da dimensão dos vetores representativos de palavras consoante o tamanho do vocabulário. Quanto maior o tamanho do vocabulário, maior tem de ser o vetor e mais memória ocupa.

2. Um vetor unitário da forma $|\psi\rangle = \sum_{i=1}^{n} \alpha_i |\psi_i\rangle$ onde todas as entradas são zeros exceto uma implica um espaço de palavras muito menor que o espaço do vocabulário. No espaço do vocabulário cada palavra é representada por $|\psi\rangle = \sum_{i=1}^{n} \alpha_i |\psi_i\rangle$. Contudo, as palavras não são independentes e, como consequência, os graus de liberdade diminuem. Isto, por sua vez, implica que os vetores $|\psi_i\rangle$ não sejam ortogonais nem bases do espaço.

Com a representação distribuída, os problemas levantados por OHE podem ser ultrapassados. Neste tipo de abordagem, cada vetor é preenchido com valores contínuos invés de valores contidos no conjunto $\{0, 1\}$. Para além disso, a sua dimensão é bastante reduzida quando comparada com o tamanho do vocabulário a representar pois a dimensão já não é o número de palavras presentes no vocabulário, mas sim o número de características escolhidas para representar as palavras. Apesar de o número de características ser escolhido e fixo em antemão, o significado de cada característica é geralmente desconhecido.

Retomando ao exemplo dado e aplicando agora uma representação distribuída, considere-se que cada palavra é representada por um vetor de dimensão igual a três ou, por outras palavras, são utilizadas três características na representação de uma palavra. Para uma maior simplificação, considere-se apenas as três palavras *past*, *present* e *distinction* cujos vetores transpostos associados se encontram presentes na tabela 2.2. Os valores presentes nesta tabela não derivam de nenhum processo, foram maioritariamente escolhidos de uma forma aleatória para fins inteiramente ilustrativos. A tabela 2.3, por sua vez, ilustra

a semelhança entre vetores através da similaridade de cosseno entre os mesmos.

Tabela 2.2: Vetorização das palavras considerando uma representação distribuída. Os valores presentes são puramente ilustrativos, servindo apenas para demonstrar a técnica e os seus benefícios.

| $\|\psi_1\rangle$ | $\|\psi_2\rangle$ | $\|\psi_3\rangle$ |
|---|---|---|
| [ 0,144 0,648 0,786 ] | [ 0,463 0,767 0,450 ] | [ 0,418 0,772 0,227 ] |

Tabela 2.3: Similaridade de cosseno entre os vetores associados às palavras *distinction*, $|\psi_1\rangle$, *past*, $|\psi_2\rangle$ e *present*, $|\psi_3\rangle$. Com uma representação distribuída percebe-se que existe uma ligação muito maior entre as palavras *past* e *present* do que entre as palavras *past* e *distinction*.

| $\cos\theta_{\|\phi\rangle,\|\psi\rangle} = \frac{\langle\phi \mid \psi\rangle}{\|\|\phi\rangle\|\,\|\|\psi\rangle\|}$ | $\|\psi_1\rangle$ | $\|\psi_2\rangle$ | $\|\psi_3\rangle$ |
|---|---|---|---|
| $\|\psi_1\rangle$ | 1,00 | 0,89 | 0,79 |
| $\|\psi_2\rangle$ | 0,89 | 1,00 | 0,98 |
| $\|\psi_3\rangle$ | 0,79 | 0,98 | 1,00 |

Os valores possíveis da similaridade de cosseno encontram-se no intervalo $[-1,1]$, onde quanto mais perto de 1 estiver o valor, mais semelhantes são os vetores. Deste modo, o valor da similaridade de cosseno entre o vetor consigo mesmo é um resultado trivial, não existe vetor mais semelhante a ele que ele próprio. Olhando para a linha do vetor $|\psi_2\rangle$, vetor associado à palavra *past*, verifica-se que a semelhança entre este vetor e o vetor $|\psi_3\rangle$, vetor associado à palavra *present*, é maior, logo mais semelhante, que a semelhança ao vetor $|\psi_2\rangle$, vetor associado à palavra *distinction*.

A representação distribuída resolve com êxito os problemas associados a OHE. Com um número reduzido de características ou dimensões, os vetores criados através desta técnica, para além de serem mais eficientes em termos computacionais, são também capazes de captar as semelhanças semânticas entre vetores. É, por isso, preferido a OHE quando o vocabulário a analisar tem um tamanho elevado.

### 2.1.4 Modelo *Skip-gram*

Proposto por Mikolov et al. [15], o modelo *Skip-gram* (SG) foi introduzido com o propósito de criar vetores de palavras de alta qualidade, isto é, vetores que representam de forma fiel uma palavra e as suas relações, a partir de conjuntos de dados com milhões de palavras. O modelo é um método eficiente de aprendizagem de representações distribuídas de vetores com capacidade de capturar de forma precisa as relações sintáticas entre as

palavras e, quando comparado com as técnicas de melhor desempenho que na altura em que foi proposto estavam disponíveis, consegue ter melhor resultados a um custo computacional muito mais pequeno [16].

O modelo SG, cuja arquitetura se encontra ilustrada na figura 2.3, tem como tarefa analisar uma frase e prever as palavras w(t+n) que se encontrem dentro de uma janela focada na palavra contexto w(t) escolhida de forma aleatória.

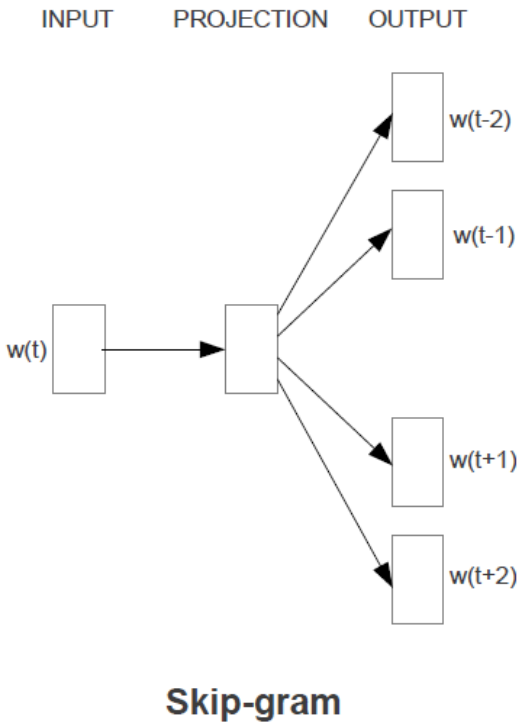


Figura 2.3: Arquitetura do modelo *Skip-gram* cujo objetivo é prever as palavras w(t+n) que se encontram dentro de uma janela, neste caso igual a dois, focada na palavra de contexto w(t). Retirado de [16].

Voltando à citação exemplo da subsecção 2.1.3, considere-se a palavra *past*, a verde, como a palavra de contexto e uma janela $j = 1$. Com esta janela, as palavras para previsão são *the* e *present*, ambas a azul.

"The distinction between the past, present and future is only a stubbornly persistent illusion."

Primeiro e para o modelo poder ser treinado e fazer previsões, a informação sobre o número de palavras no vocabulário, $V$, tem de ser fornecida ao modelo para ele poder criar uma matriz de *word embeddings* ou vetores representantes de palavras. Com uma dimensão $D \times V$, em que $D$ é a dimensão dos *word embeddings*, esta matriz é responsável por guardar todos os *word embeddings* das palavras do vocabulário e inicializada de uma forma aleatória. Designe-se a matriz por $\rho$, ilustrada na eq. 2.14.

$$\begin{bmatrix} \rho_{11} & \cdots & \rho_{1V} \\ \vdots & \ddots & \vdots \\ \rho_{D1} & \cdots & \rho_{DV} \end{bmatrix}_{D\times V} \tag{2.14}$$

O modelo SG obtém o *word embedding* da palavra *past* multiplicando o seu vetor na forma OHE, $|\theta_{past}\rangle$, pela matriz $\rho$. Cada coluna da matriz $\rho$ representa uma palavra do vocabulário e a multiplicação das matrizes, eq. 2.15, resulta da seleção da coluna associada à palavra *past*, ou seja, num vetor de dimensão $D$ cuja designação será $|\psi_{past}\rangle$.

$$\overset{\rho}{\begin{bmatrix} \rho_{11} & \cdots & \rho_{1V} \\ \vdots & \ddots & \vdots \\ \rho_{D1} & \cdots & \rho_{DV} \end{bmatrix}_{D\times V}} \times \overset{|\theta_{past}\rangle}{\begin{bmatrix} 0 \\ \vdots \\ \theta_{100} = 1 \\ \vdots \\ 0 \end{bmatrix}_{V\times 1}} = \overset{|\psi_{past}\rangle}{\begin{bmatrix} \rho_{1\,100} \\ \vdots \\ \rho_{D\,100} \end{bmatrix}_{D\times 1}} \tag{2.15}$$

O vetor $|\psi_{past}\rangle$ é um vetor pertencente ao espaço de Hilbert $\mathbb{R}^D$. Cada posição dentro do vetor guarda o valor real associado a cada uma das $D$ características que o modelo utiliza na representação distribuída da palavra.

De seguida, multiplica-se o vetor $|\psi_{past}\rangle$ pela transposta da matriz $\rho$. O resultado é um vetor $|\phi\rangle$ de dimensão $V \times 1$ que em cada posição guarda o resultado do produto interno entre o vetor da palavra contexto *past* e o vetor de uma palavra do vocabulário, eq. 2.16.

$$\overset{\rho^T}{\begin{bmatrix} \rho_{11} & \cdots & \rho_{1D} \\ \vdots & \ddots & \vdots \\ \rho_{V1} & \cdots & \rho_{VD} \end{bmatrix}_{V\times D}} \times \overset{|\psi_{past}\rangle}{\begin{bmatrix} \rho_{1\,100} \\ \vdots \\ \rho_{D\,100} \end{bmatrix}_{D\times 1}} = \overset{|\phi\rangle}{\begin{bmatrix} \langle\psi_1 \mid \psi_{past}\rangle \\ \vdots \\ \langle\psi_V \mid \psi_{past}\rangle \end{bmatrix}_{V\times 1}} \tag{2.16}$$

Quando o modelo foi inicialmente proposto, o passo seguinte era fornecer o valor de cada posição à função *softmax* do modelo, eq. 2.17, e calcular a probabilidade condicionada de a palavra $n$ estar presente na janela de previsão sabendo que a palavra *past* é a palavra de contexto [16]. O problema desta abordagem encontrava-se na necessidade de percorrer todas as palavras do vocabulário para o cálculo da probabilidade, o que num vocabulário de grandes dimensões implicava um grande custo computacional.

$$p(|\psi_n\rangle \mid |\psi_{past}\rangle) = \frac{\exp\left(\langle\psi_n \mid \psi_{past}\rangle\right)}{\sum_{v=1}^{V} \exp\left(\langle\psi_v \mid \psi_{past}\rangle\right)} \tag{2.17}$$

Uma das melhorias introduzidas por Mikolov et al. [16] para melhorar o tempo de cálculo foi usar o processo de amostragem negativa. O modelo invés de utilizar a função *softmax* da eq. 2.17, utiliza uma função sigmoide no cálculo da probabilidade e aplica uma regressão logística. Tendo em conta a palavra contexto *past*, é escolhida uma palavra que esteja dentro da janela, por exemplo *present*, e $k$ palavras aleatórias que estejam fora da janela. O produto interno entre a palavra contexto e as outras é fornecido à função sigmoide que, por sua vez, devolve as probabilidades associadas. Os restantes cálculos são realizados de forma a garantir que a probabilidade associada à palavra *present* seja maior que qualquer probabilidade associada às outras $k$ palavras. Ao contrário da função *softmax* que percorre o vocabulário todo, a amostragem negativa apenas percorre $k+1$ palavras o que permite ao modelo ter um treino muito mais rápido. Para além desta mudança, foi introduzida uma outra, a subamostragem de palavras frequentes, eq. 2.18, onde cada palavra utilizada no treino do modelo tem uma probabilidade $P(|\psi_n\rangle)$ de ser descartada baseada na sua frequência de ocorrência $f(|\psi_n\rangle)$ e num limite $t$.

$$P(|\psi_n\rangle) = 1 - \sqrt{\frac{t}{f(|\psi_n\rangle)}} \tag{2.18}$$

Após os resultados das probabilidades, são realizados mais alguns cálculos e, através de retropropagação, a matriz $\rho$ é atualizada. No final do treino, os valores de $\rho$ já permitem obter *word embeddings* de grande qualidade e com uma boa capacidade de representação das relações entre palavras.

### 2.1.5 Representação Gráfica: *t-Distributed Stochastic Neighbor Embedding*

A visualização dos resultados de modelos como o SG requer sempre a utilização de alguma técnica de redução de dimensões devido aos vetores criados pelos modelos serem de dimensões elevadas e para além da capacidade humana de visualização.

Qualquer tipo de técnica de redução de dimensões, com o intuito de facilitar a visualização por humanos, tem como foco maximizar a preservação estrutural da informação inicial quando a reduz para duas ou três dimensões. Porém, diferentes técnicas preservam diferentes estruturas. As técnicas mais tradicionalmente usadas, como a análise das componentes principais, são métodos lineares cujo objetivo é manter os pontos dissimilares o mais afastados possíveis na sua representação de dimensões reduzidas. Contudo, a visualização de determinados casos, como em PLN, é muito mais útil manter os pontos semelhantes mais próximos do que os pontos menos semelhantes mais afastados, o que habitualmente não é possível com métodos lineares [17].

Uma das técnicas úteis para os casos onde a estrutura local é importante é a técnica *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Desenvolvida por Maaten e Hinton [17], t-SNE é uma variação do método *Stochastic Neighbor Embedding* (SNE) mais fácil de otimizar e capaz de produzir melhores visualizações onde não se verifica com tanta

frequência a tendência de agrupamento de pontos no centro do mapa criado.

A técnica original SNE, num primeiro estágio, converte as distâncias euclidianas entre dois pontos no espaço de maiores dimensões em probabilidades condicionais. A similaridade entre os pontos $x_j$ e $x_i$ é a probabilidade condicional, $p_{j\,|\,i}$, eq. 2.19, que $x_j$ tem de escolher $x_i$ como vizinho se os vizinhos forem escolhidos tendo em conta a sua densidade de probabilidade debaixo de uma gaussiana centrada em $x_i$ e variância $\sigma_i^2$. Pontos que estejam próximos apresentam uma $p_{j\,|\,i}$ alta, enquanto que pontos mais afastados apresentam uma $p_{j\,|\,i}$ mais baixa. Uma probabilidade análoga é definida para o espaço de menores dimensões, eq. 2.20.

$$p_{j\,|\,i} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \tag{2.19}$$

$$q_{j\,|\,i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)} \tag{2.20}$$

A partir das duas probabilidades, SNE tenta encontrar os pontos no espaço de menores dimensões que minimizem a diferença entre as probabilidades e fá-lo considerando uma função custo não simétrica. Devido à sua falta de simetria, erros diferentes são ponderados com pesos diferentes. Em particular, SNE tenta reter a estrutura local dos dados penalizando muito mais os erros associados à representação de pontos próximos por pontos muito afastados e penalizando menos os erros de representação de pontos muito afastados por pontos próximos. As visualizações resultantes são, geralmente, razoavelmente boas, no entanto sofrem de problemas de aglomeração e a otimização da técnica não é fácil de se fazer. O t-SNE surge como uma solução para esses problemas.

As principais diferenças entre SNE e t-SNE encontram-se na sua função custo e tipo de distribuição utilizada nos cálculos da similaridade no espaço de baixas dimensões. A nova função custo, para além de ser simétrica, é mais simples e mais rápida de calcular, sendo capaz de produzir resultados tão bons ou melhores que os resultados de SNE. Em relação à distribuição utilizada, a distribuição normal continua a ser utilizada nos cálculos no espaço de elevadas dimensões mas é substituída por uma distribuição *t-student* nos cálculos referentes ao espaço de baixas dimensões, melhorando os problemas de aglomeração. Com estas duas mudanças, t-SNE consegue obter melhores resultados e mais rápido.

## 2.2 Espaço das Palavras

O espaço das palavras é um espaço pouco habitual no sentido em que não partilha nenhuma das características dos espaços commumente utilizados. A título de explicação,

comece-se por ilustrar o que geralmente se usa.

Considere-se uma grelha de 4 x 4, onde cada zona pode conter até 5 objetos diferentes, tal como na figura 2.4. O número associado a cada zona é o número de objectos que nela se encontra.

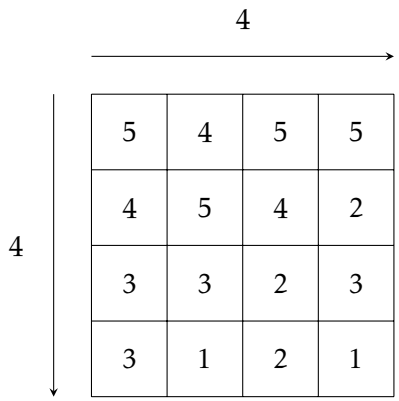


Figura 2.4: Exemplo de um grelha 4 x 4 onde cada zona pode conter até 5 objectos diferentes. Cada zona encontra-se preenchida com o número de objetos nela presente.

Considere-se agora que a grelha representa uma área de 400 m x 400 m e cada zona contém a informação sobre quantos prédios estão presentes dentro da sua área. Através de uma pequena análise, consegue-se retirar algumas conclusões. Analise-se, por exemplo, a densidade. Se se colorir as áreas mais densamente populadas com um cizento escuro e as áreas menos populadas com um cinzento claro, o resultado é igual ao da figura 2.5. Daqui conclui-se que o canto superior esquerdo é mais densamente populado que o canto inferior direito. Uma outra informação que se pode recolher é a evolução da densidade ao longo do tempo através da criação de um gráfico que ilustre, por exemplo, a mudança da densidade das várias zonas com o tempo.

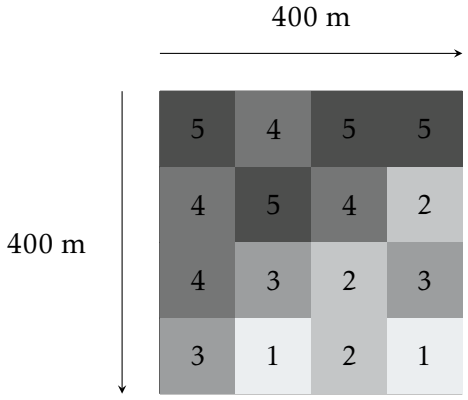


Figura 2.5: Grelha representante de uma área de 400 m x 400 m. Cada zona encontra-se colorida de acordo com o número de prédios presentes na área. Ao cinzento mais escuro corresponde uma zona mais populada e ao cinzento claro uma zona menos populada.

Contudo, independentemente do tipo de análise que se realiza, em nenhum momento se questiona como é o espaço onde se colocam os dados. Assume-se que todas as zonas são iguais e que os objetos são independentes uns dos outros, que a adição ou remoção de um objeto de uma zona da grelha não afeta os objetos presentes numa outra zona. Por outras palavras, assume-se que o espaço é plano.

E se os objetos forem dependentes uns dos outros? Imagine-se a mesma grelha mas com os objetos ligados aos objetos adjacentes como ilustrado na figura 2.6 e que o número de ligações realizado por cada objeto não é constante de objeto para objeto. Assim é o caso das palavras. A dependência observada entre palavras altera drasticamente a forma como o problema passa a ser analisado. As palavras e o texto são algo que não existe na natureza, sendo natural apenas ao ser humano e existindo unicamente na sua mente. As palavras não existem num espaço de "fundo", as palavras são o próprio espaço. Ter uma grelha 4 x 4 ou qualquer outra deixa de ter significado ou, pelo menos, torna-se difícil de interpretar porque se desconhece o espaço onde se encontra a grelha. Por outro lado, as próprias palavras e o seu significado também alteram este espaço. O significado de uma palavra está dependente das palavras do seu contexto. Uma palavra num dicionário vem sempre acompanhada de vários significados consoante a frase em que se encontra e o pensamento de uma palavra invoca sempre o pensamento de outras. A relação intrínseca entre palavras é tão forte que alterar essa relação implica alterar as palavras em si, à semelhança de quando se trata de um fenómeno de entrelaçamento quântico. Tendo em conta estas duas características, o espaço das palavras deixa de ser uniforme. Cada palavra encontra-se relacionada com várias outras palavras, umas com mais ligações, outras com menos e observa-se um fenómeno de entrelaçamento em grande escala onde mudar uma palavra implica mudar todo o espaço. Esta interligação de palavras e inconsistência de ligações pode ser visualizada na figura 2.7 onde se encontra ilustrada uma *scale-free network* criada com base nas descrições dos filmes de crime [18, 19]. Na figura facilmente se verifica que existem muitas palavras a fazer poucas ligações e poucas palavras a fazer muitas ligações. A grande consequência de trabalhar num espaço cujos objetos não são independentes e se encontram todos interligados é o espaço de estudo não ser "Plano".

O primeiro problema derivado do facto de se ter de trabalhar num espaço não plano é a impossibilidade de utilizar a matemática habitual. A matemática comum está adaptada a espaços planos e deixa de funcionar em espaços de geometrias para além dessa. Sem a matemática não é possível realizar nenhuma análise e o estudo das palavras torna-se impossível. Um outro problema dos espaços planos é a inabilidade das RNAs mais simples conseguirem trabalhar com eles. Os algoritmos destas redes estão feitos de forma a procurarem padrões nos objetos assumindo que os mesmo são independentes o que, como já foi explicado, não se verifica para as palavras.[1] Assim, a solução encontrada para

[1] Para além disso, o estudo de variedades matemáticas está fora do âmbito deste trabalho por não fazer

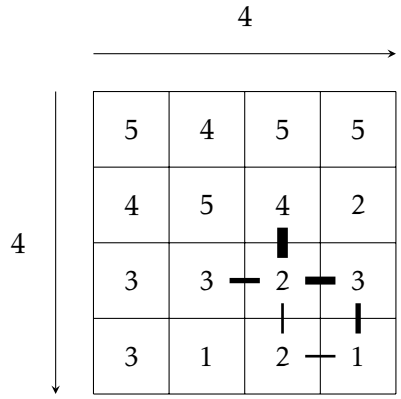


Figura 2.6: Grelha 4 x 4 onde cada zona pode conter até 5 objectos diferentes, podendo cada um ligar-se aos objetos adjacentes. Quanto mais grossa a linha da ligação mais ligações existem entre os objetos das duas áreas.

ultrapassar todos estes obstáculos foi a realização de uma projeção do espaço para um espaço plano, nomeadamente, um espaço de Hilbert.

A projeção para o espaço de Hilbert é feita garantindo que o novo espaço, agora plano, é de menores dimensões que o espaço original. Esta diminuição de dimensões obriga a que haja uma redução da informação a guardar, ficando guardada apenas a informação mais crucial. Explicando de uma outra forma, reduzir a dimensionalidade e garantir que todas as palavras são guardadas requer encontrar as características mais significativas e guardar as palavras tendo em conta a maneira como essas características se expressam na palavra a ser projetada. Dentro do espaço de Hilbert, estas características são os vetores base do espaço. Porém, com esta abordagem levanta-se um outro problema que é como aceder às representações das palavras dentro do espaço de Hilbert se os vetores base do mesmo não são conhecidos? A resposta encontra-se no modelo SG. O modo de operação do modelo permite chegar à importante matriz densidade sem ter de fazer pressupostos sobre os vetores base. Esta matriz, por sua vez, permite chegar às representações das palavras dentro do espaço.

Dentro do modelo SG, a camada de *input* é o espaço inicial das palavras, figura 2.8. Cada unidade da camada de *input* representa um vetor $|\psi_i\rangle$ com $i \in [1, V]$ utilizado na representação de cada uma das $V$ palavras do vocabulário. Por sua vez, a representação das palavras dentro deste espaço é realizada através de um vetor unitário da forma $|\psi\rangle = \sum_{i=1}^{V} \alpha_i |\psi_i\rangle$ com todas as entradas zero exceto uma. Como as palavras não são independentes, os vetores $|\psi_i\rangle$ não são ortogonais e não formam uma base do espaço das palavras, tornando este espaço muito menor que o espaço do vocabulário. É na passagem da camada de *input* para a camada oculta que se realiza a projeção para o espaço de Hilbert de $D$ dimensões e que os vetores $|\psi_i\rangle$ se transformam nos vetores $|\phi_j\rangle$ com

parte do currículo de Engenharia Física.

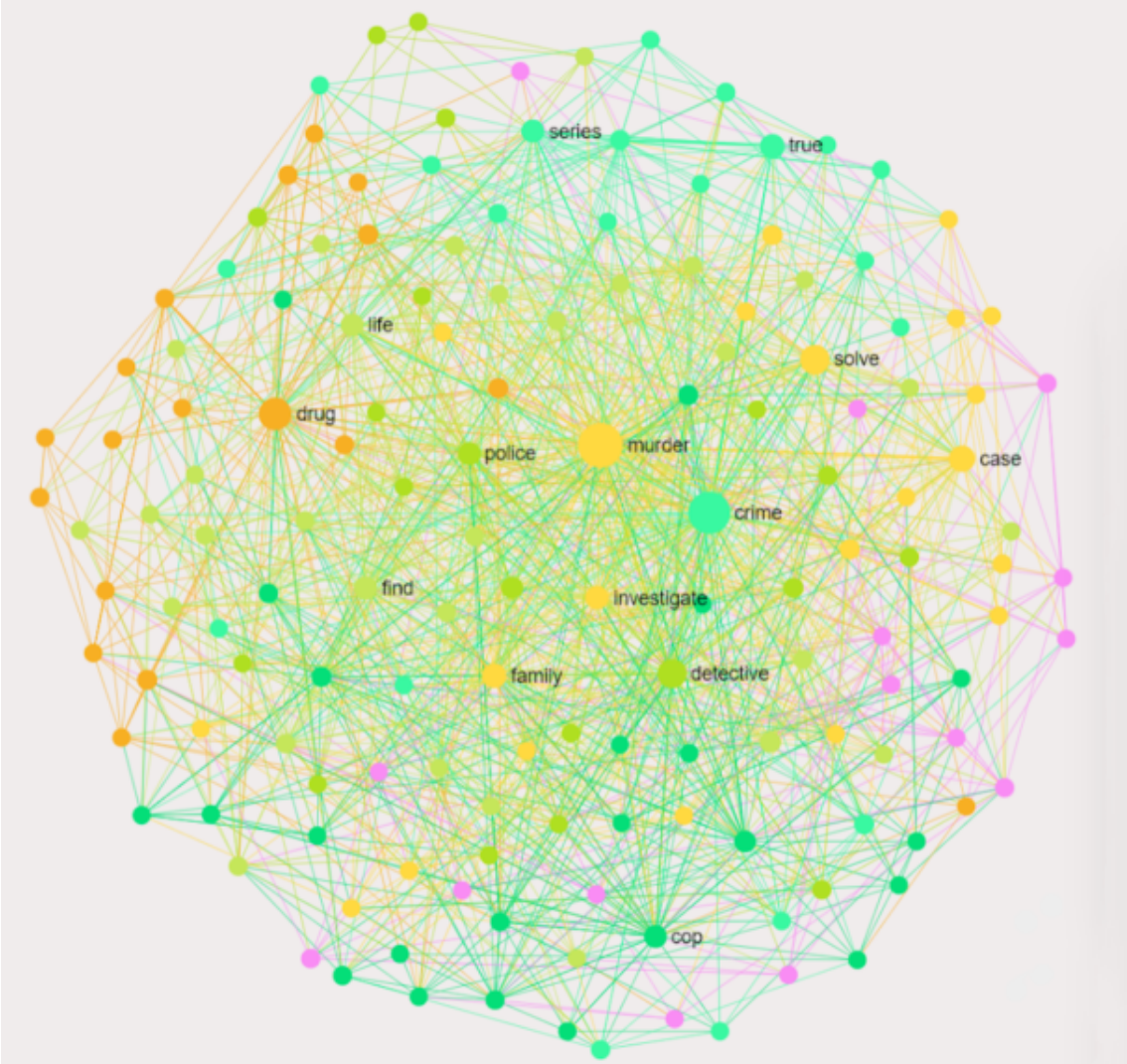


Figura 2.7: *Scale-free network* aplicada ao texto. A *scale-free network* foi criada com base nas descrições dos filmes de crime. Adaptada de [19].

$j \in [1, D]$. As palavras passam a ser representadas por $|\psi\rangle = \sum_{i=1}^{D} \beta_j |\phi_j\rangle$ onde cada $|\phi_j\rangle$ é agora ortogonal aos outros e uma base do espaço de Hilbert. A projeção é feita garantindo que aquando no processo de descodificação da palavra dentro do espaço de Hilbert, figura 2.9, se obtenham as palavras que a rodeiam o que, por sua vez, garante que os vetores $|\phi_j\rangle$ sejam ortogonais. Esta projeção é feita pela matriz $\rho$ referida na subsecção 2.1.4 na eq. 2.14 e aqui na eq. 2.21.

$$\begin{bmatrix} \rho_{11} & \cdots & \rho_{1V} \\ \vdots & \ddots & \vdots \\ \rho_{D1} & \cdots & \rho_{DV} \end{bmatrix}_{D \times V} \tag{2.21}$$

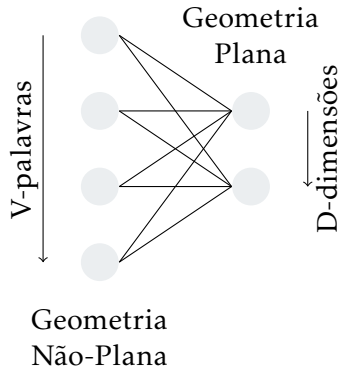


Figura 2.8: Codificação da camada de *input* para a camada oculta da rede. A camada de *input* engloba as V palavras e o seu espaço enquanto que a camada oculta engloba a projeção das palavras num espaço de Hilbert de D dimensões. A projeção de um espaço para o outro permite passar de uma geometria não-plana para um espaço plano.

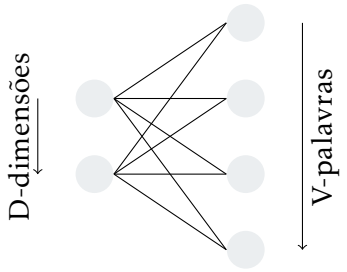


Figura 2.9: Descodificação da camada oculta para a camada de *output*.

A matriz é inicializada com valores arbitrários e, após o treino do modelo é, efetivamente, a matriz densidade responsável pela codificação dos vetores de palavras no espaço inicial para o espaço de Hilbert. A matriz, em cada coluna, guarda um vetor de $D$ dimensões onde cada entrada se traduz na relação da palavra com uma das $D$ características encontradas, sendo estas características nada mais que as palavras que tipicamente a rodeiam.

Capítulo 3

# Metodologia e Origem dos Dados

Neste capítulo é apresentada a origem dos dados utilizados para gerar os mapas de palavras assim como os passos seguidos para a obtenção dos resultados finais. Porém, em nenhuma ocasião ao longo do capítulo é apresentado o código associado aos resultados na sua íntegra, estando esse presente num repositório do *GitHub* [20]. Quase todo o código utilizado foi desenvolvido neste trabalho, exceto algumas porções que foram adaptadas do trabalho de Bernardo [21]. Contudo, tanto nos próprios ficheiros como nas legendas, o código adaptado de Bernardo [21] encontra-se devidamente identificado.

## 3.1 Origem dos Dados

Entenda-se por *corpus* o conjunto de textos recolhidos para o estudo da densidade do espaço de palavras. Os *corpora* utilizados no estudo têm duas composições diferentes. Designado por *corpus* 1, este é constituído por textos de teor económico e financeiro recolhidos de diferentes fontes e provenientes de várias épocas temporais. A sua composição é variável e textos que dela fazem parte dependem do intervalo de tempo a analisar. Por outro lado, o *corpus* 2 tem uma composição estável, sendo constituído por alguns dos textos presentes no *corpus* 1. Porém, os textos em questão não formam nenhum intervalo específico, sendo apenas utilizados para gerar um *corpus* de grande dimensão.

### 3.1.1 *Corpus* 1

O *corpus* 1 é um *corpus* feito à medida do intervalo de estudo. Por exemplo, o estudo do intervalo entre 2000 e 2004 requer um *corpus* diferente do *corpus* para um estudo entre 2000 e 2008. Contudo, os *corpora* não são totalmente diferentes, estando o primeiro *corpus* contido no segundo. A informação possível se ser encontrada dentro dos mesmos

encontra-se na tabela 3.1.

Tabela 3.1: Descrição dos relatórios e notícias utilizados na formação do *corpus* 1 e respetivos intervalos temporais.

| Nome | Tipo de Dados | Período |
|---|---|---|
| **Relatórios Anuais do Banco Central Europeu** | Relatórios | 1991 - 2019 |
| **Boletins Económicos** | Relatórios | 2015 - 2020 |
| **Previsões Económicas da Comissão Europeia** | Relatórios | 1998 - 2019 |
| **Perspetivas Económicas Regionais - Europa do Fundo Monetário Internacional** | Relatórios | 2007 - 2009; 2018 |
| ***Financial News Articles* de *Webhose*** | Notícias | Julho - Outubro de 2015 |
| ***Financial News Unprocessed*** | Notícias | Junho - Julho de 2017 |
| ***Reuters* - 21578** | Notícias | 1987 |

Proveniente do trabalho de Bernardo [21], o *corpus* 1 é um *corpus* muito focado na situação económica e financeira da Europa e da União Europeia ao longo dos períodos reportados na tabela 3.1. O seu objetivo é perceber a evolução do *corpus* e das relações entre palavras ao longo do tempo por isso os relatórios e as notícias recolhidas compreendem um vasto período entre 1987 a 2020.

Relatórios:

- Os Relatórios Anuais do Banco Central Europeu descrevem as atividades do Sistema Europeu de Bancos Centrais e políticas monetárias do Eurosistema. Os relatórios recolhidos vão desde 1991 a 2019 [22].

- Os Boletins Económicos contêm a informação nas quais se baseiam as decisões sobre a política monetária do Banco Central Europeu. Os boletins enriquecem o intervalo de anos entre 2015 e 2020 [23].

- As previsões económicas lançadas pelas Comissão Europeia descrevem as perpetivas de evolução de alguns indicadores económicos [24]. Para além disso, incluem também informação económica sobre os países membros, futuros membros e economias mundiais com capacidade de afetar a União Europeia e a zona Euro [21].

- As Perspetivas Económicas Regionais publicadas pelo Fundo Monetário Internacional discutem as perspetivas e os desenvolvimentos mais recentes nos países de várias regiões. Em adição, discutem também o desempenho das políticas económicas e os desafios associados, fornecendo informação específica sobre cada país. Os relatórios recolhidos focam-se na região da Europa e vão desde 2007 a 2009, havendo um relatório de 2018 [25].

Notícias:

- O conjunto de dados *Financial News Articles*, recolhido de forma livre de *Webhose* [26], contém notícias de teor económico provenientes de várias agências internacionais lançadas no ano de 2015, mais especificamente, lançadas entre Julho e Outubro.
- O conjunto de notícias *Financial News Unprocessed*, disponível para uso público [27], contém notícias escritas por *Reuters* e *The Wall Street Journal* entre Junho e Julho de 2017.
- O conjunto de dados de acesso público composto por 10788 artigos de notícias publicadas por *Reuters* em 1987 [28].

Todos os textos recolhidos e a sua distribuição temporal encontram-se resumidos de forma sucinta na figura 3.1. Apesar de existirem alguns intervalos sem quaisquer dados recolhidos, os dados disponíveis cobrem aproximadamente três décadas da economia europeia, permitindo um bom estudo da evolução da relação entre palavras ao longo desse tempo.

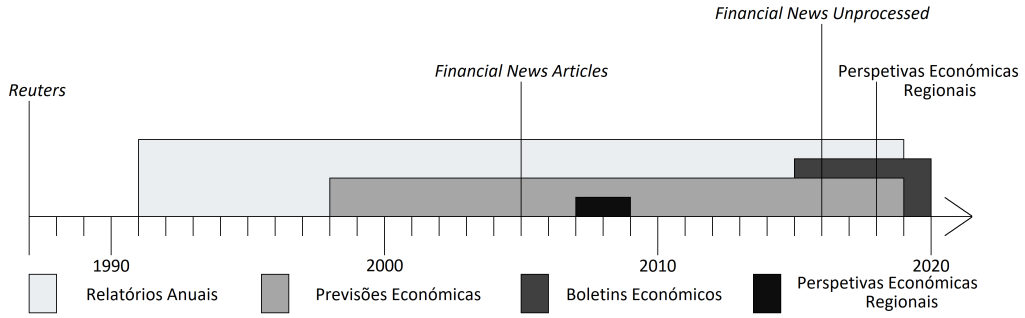


Figura 3.1: Distribuição temporal dos textos utilizados na criação do *corpus* 1.

### 3.1.2 *Corpus* 2

O *corpus* 2 é composto por um conjunto fixo de textos pertencentes ao *corpus* 1. Ao contrário desse *corpus* onde o obejtivo se prende com o estudo da evolução das palavras ao longo do tempo, com o *corpus* 2 pretende-se estudar o espaço das palavras quando o *corpus* é de grandes dimensões. O *corpus* 2 é composto por:

- As notícias de *Financial News Articles*.

- As notícias de *Financial News Unprocessed*, do mês de Junho de 2017.
- Os relatórios do Fundo Monetário Internacional.

A dimensão do *corpus* é, no total, cerca de catorze milhões de palavras, tendo, aproximadamente, 4.8 x $10^5$ palavras diferentes.

## 3.2 Pré-Processamento do *Corpus*

Antes do *corpus* ser fornecido ao modelo para treino é necessário um pré-processamento do mesmo. O pré-processamento do *corpus* garante que o mesmo se encontra preparado para a análise a realizar e que os modelos criados não contêm palavras irrelevantes.

### 3.2.1 Organização do *Corpus* 1

Para uma melhor organização e fácil acesso aos textos associados ao *corpus* 1, estes encontram-se divididos em pastas com uma hierarquia igual à hierarquia da figura 3.2. A pasta *Years* contém todas as pastas referentes ao anos presentes no intervalo global de anos e cada uma delas inclui três outras pastas e um ficheiro. Uma das pastas, *UNPROCESSED_TEXT*, contém as notícias e relatórios desse período com a sua formatação original, outra, *CLEAN_TEXTS*, contém esses mesmos textos já processados e limpos e outra, *REMOVED_NAMES*, contém os ficheiros de palavras removidas no processo de limpeza. O ficheiro é o *corpus* principal, *CLEAN_MAIN_CORPUS*, composto por todos os *corpora* individuais da pasta *CLEAN_TEXTS*.

Este tipo de hierarquia permite não só ver que *corpus* estão em que ano e em que quantidade mas também muito mais facilmente adicionar novos *corpus* a cada ano, sendo apenas necessário limpar os *corpora* acrescentados e criar um novo *corpus* principal. Contudo, a abordagem utilizada ao longo do trabalho foi ligeiramente diferente desta. Invés de se limpar os *corpora* individuais e de se juntar tudo num grande *corpus*, primeiro faz-se a junção dos *corpora* e só depois se realiza a sua limpeza. Neste caso, juntar novos dados obriga a limpeza do novo *corpus* principal, um processo que, para além de processar dados já processados, é muito mais demorado do que a limpeza de *corpora* mais pequenos. Em ambas as abordagens os resultados são iguais mas a mais eficiente permite saltar a barreira de tempo caso, no futuro, se pretenda adicionar mais textos a um *corpus* já formado e processado. Assim, todo o código utilizado sofreu pequenas alterações para se adaptar a esta hierarquia.

### 3.2.2 Limpeza do *Corpus*

A limpeza do *corpus* é essencial à criação de um modelo de qualidade. O código utilizado na limpeza do *corpus* foi desenvolvido por Bernardo [21] e alterado para acomodar a flexibilidade do *corpus* 1. A íntegra do código encontra-se num repositório do *GitHub* [20]

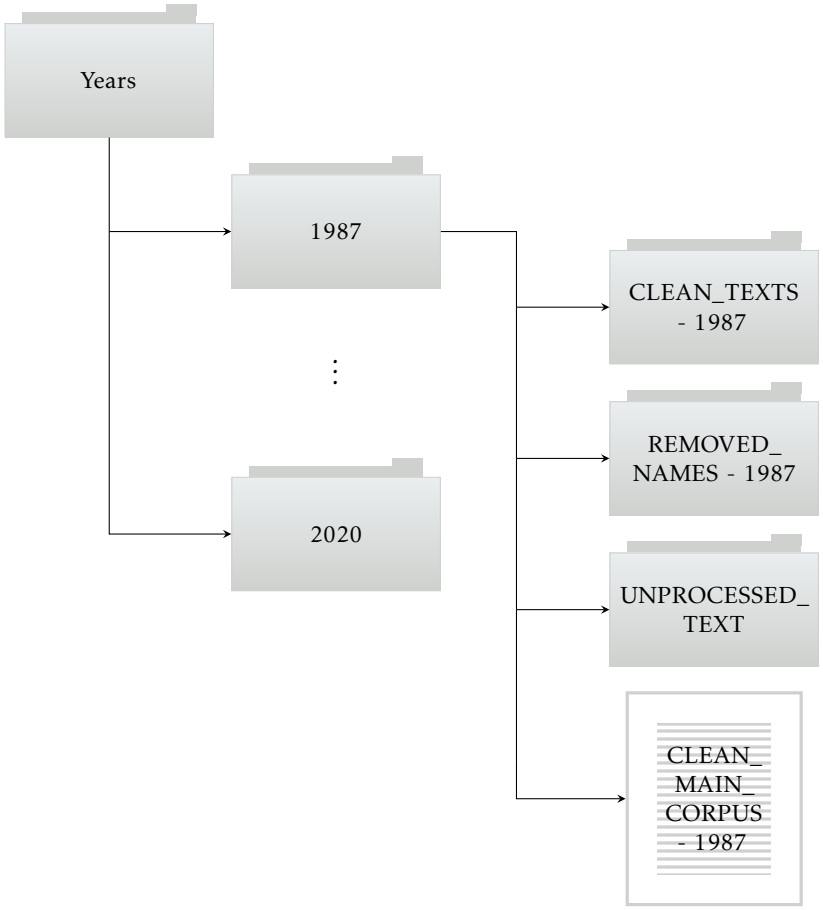


Figura 3.2: Hierarquia de pastas associadas aos textos do *corpus* 1.

e a explicação completa do código original no trabalho de Bernardo [21].

A limpeza do *corpus* tem três grandes funções:

- Eliminar os caracteres não alfabéticos: quaisquer carateres não alfabéticos, como números, pontuação ou símbolos, não carregam significado dentro do *corpus*.
- Eliminar as *stopwords*: as *stopwords* são palavras que surgem frequentemente na linguagem natural, ajudando na coesão do texto. Porém, não contribuem para o valor semântico do texto. Alguns exemplos são as palavras *and* e *or*.
- Simplificar o *corpus*: a simplificação das palavras presentes no *corpus* permite reduzir de forma considerável o seu tamanho. Um *corpus* mais pequeno implica um menor tempo de processamento na etapa de treino do modelo e uma maior precisão do modelo final. A simplificação do *corpus* é feita através da caracterização da morfologia de cada palavra e consequente redução da mesma à sua forma mais simples. Por exemplo, os nomes são todos colocados no seu singular, os verbos no seu infinitivo e os adjetivos no seu grau normal. A este processo chama-se lematização.

No trabalho original de Bernardo [21], os dados encontravam-se separados em três grupos: o *corpus* antes de 2000, o *corpus* de 2000 a 2008 e o *corpus* após 2008. Apesar de ser um processo tedioso e demorado, a limpeza do *corpus* podia ser feita fornecendo manualmente o *corpus* um a um, visto o número de ficheiros ser reduzido e o processo só ter de ser realizado uma vez. O mesmo processo se pode aplicar ao *corpus* 2 porque a sua composição se mantém sempre constante.

Contudo, tratamento do *corpus* 1 requer fazer umas alterações no código de Bernardo [21]. Limpar o *corpus* 1 sempre que a sua composição muda não é prático porque o tempo de processamento, para além de longo, é proporcional ao tamanho do *corpus*. A primeira alteração é a transformação de parte do código de limpeza na função `corpusCleaning` cujo argumento é um dicionário. Invés do *corpus* a processar ser fornecido pelo utilizador, os vários *corpora* são importados recorrendo às listas guardadas no dicionário. O corpo inicial da função encontra-se na listagem 3.1.

```
1  def corpusCleaning(unpCorpusDict):
2      # Cleans the corpus by eliminating non-alphabetic characters, stopwords
       and characters smaller than 2
3      # unpCorpusDict - dictionary whose keys are the years to be processed and
       the values are the unprocessed corpora list
4      # return        - none
5
6      # Loop for going throught the unprocessed corpus dict
7      for key in unpCorpusDict.keys():
8
9          year = key
10         unpCorpusList = unpCorpusDict[key]
11
12         for corpus in unpCorpusList:
13
14             # Path to find corpus
15             corpusPath = os.path.join(os.getcwd(), "Years", str(year), "
       UNPROCESSED_TEXT", corpus)
16
17             # Open corpus
18             file = open(corpusPath, encoding='latin-1')
19             text = file.read()
20             file.close()
21
22             (...)
```

Listagem 3.1: Função `corpusCleaning`, responsável pela limpeza dos *corpora*. Adaptado de [21].

A segunda alteração encontra-se na forma como os *corpora* processados e limpos são guardados. Todos os *corpora* limpos são guardados dentro da pasta *CLEAN_TEXTS*, enquanto que os ficheiros associados aos nomes removidos na limpeza são guardados na

pasta *REMOVED_NAMES*, ambas presentes na pasta do ano a que correspondem e criadas no momento de gravação do primeiro *corpus* limpo. Este processo de gravação encontra-se ilustrado na listagem 3.2.

```
(...)
# Folders to save clean corpus and removed names
saveFolder = os.path.join(os.getcwd(), "Years", str(year), "CLEAN_TEXTS - " +
    str(year))
removedFolder = os.path.join(os.getcwd(), "Years", str(year), "REMOVED_NAMES -
     " + str(year))

# Check if folders exist, if not create them
if (not os.path.exists(saveFolder)):
    os.makedirs(saveFolder)

if (not os.path.exists(removedFolder)):
    os.makedirs(removedFolder)

# Save clean corpus
with open(os.path.join(saveFolder, "CLEAN_TEXT - " + corpus), "w", encoding='
    utf-8') as txt_file:
    for text in text_:
        txt_file.write("".join(text) + " ")

# Save removed names txt
with open(os.path.join(removedFolder, "REMOVED_NAMES - " + corpus), "w",
    encoding='utf-8') as txt_file:
    for row in trash:
        txt_file.write("".join(row) + " ")
(...)
```

Listagem 3.2: Gravação dos *corpora* já limpos. Proveniente da função `corpusCleaning`. Adaptado de [21].

### 3.2.3 Formação do *Corpus* Principal

Com todos os *corpora* individuais e de cada ano já processados e limpos, a formação do *corpus* principal é o passo seguinte. O *corpus* principal é importante na medida em que facilita o acesso aos *corpus* individuais para a posterior criação do *corpus* 1, combinando-os num só ficheiro denominado *CLEAN_MAIN_CORPUS*. A função responsável pela sua formação designa-se por `corpusCombinatorProcessed` presente na listagem 3.3.

```
def corpusCombinatorProcessed(year1, year2):
    # Combines all processed corpora present in the folder "CLEAN_TEXTS - YEAR
    " that's inside the folder of each year into one main clean corpus called "
    CLEAN_MAIN_CORPUS - YEAR.txt"
    # year1  - first year of the interval
```

```
4     # year2  - last year of the interval
5     # return - none
6 (...)
```

Listagem 3.3: Função `corpusCombinatorProcessed`, responsável pela aglomeração dos *corpora* processados num ficheiro principal. Desenvolvido neste trabalho.

Fornecendo o intervalo de trabalho à função, cada ano desse período é percorrido e todos os *corpora* processados são recolhidos. Com o auxílio de um ciclo, cada *corpus* é lido e copiado para o fim do *corpus* principal *CLEAN_MAIN_CORPUS* guardado em cada ano, listagem 3.4.

```
1 (...)
2 # Loop for reading the individual corpus and appending it to the main corpus
3 for corpus in corpusFiles:
4     file = open(os.path.join(yearFolder, "CLEAN_TEXTS - " + str(year), corpus)
      , "r", encoding = "utf-8")
5     text = file.read()
6     file.close()
7
8     with open(mainCorpus, "a", encoding = "utf-8") as file:
9         file.write("".join(text) + " ")
10 (...)
```

Listagem 3.4: Ciclo responsável pela leitura e cópia dos *corpora* individuais para o *corpus* principal. Proveniente da função `corpusCombinatorProcessed`. Desenvolvido neste trabalho.

### 3.2.4 Formação *Corpus* 1

A formação do *corpus* 1 é a última etapa dentro do pré-processamento do *corpus*. Ao contrário das etapas anteriores, o *corpus* 1 é formado à medida que os dados vão sendo gerados e está sempre em constante atualização. Por exemplo, considere-se que se pretende gerar mapas de palavras para o intervalo entre 2010 e 2020 de 2 em 2 anos. Para cada sub-intervalo existe um *corpus* que é composto por todos os dados dos anos desse período assim como os dados provenientes dos sub-intervalos anteriores. Com o avanço para o sub-intervalo seguinte, os dados deste novo intervalo são adicionados ao *corpus* existente e este é de novo analisado.

A função responsável por essa atualização designa-se por `corpusCombinatorMain` e encontra-se na listagem 3.5. Considerando o intervalo de trabalho, a função percorre todos os anos verificando a existência de *corpora* principais. Caso estes existam, o seu conteúdo é adicionado ao *corpus* 1. Em cada estágio, o seu estado é guardado. Contudo e como o *corpus* 1 é formado apenas no momento de criação de dados, este não é guardado no local habitual, sendo guardado dentro da pasta responsável por também guardar os dados que se criam

a partir do *corpus* 1.

```
def corpusCombinatorMain(beginning, end, saveFolderName):
    # Combines the clean main corpora from multiple years into one main corpus
     "CORPUS.txt". If "CORPUS.txt" is already created, it appends the new
    corpora to it
    # beginning      - first year of the interval
    # end            - last year of the interval
    # saveFolderName - folder where to save and retrieve data
    # return         - "CORPUS.txt" path

    (...)

    while (year != end + 1):
        # Check if "CLEAN_MAIN_CORPUS" file exists and added it to the main
    corpus
        if os.path.exists(os.path.join(mainFolder, str(year), "
    CLEAN_MAIN_CORPUS - " + str(year) + ".txt")):
            corpus = os.path.join(mainFolder, str(year), "CLEAN_MAIN_CORPUS -
    " + str(year) + ".txt")
            file = open(corpus, "r", encoding = "utf-8")
            text = file.read()
            file.close()

            with open(mainCorpus, "a", encoding = "utf-8") as file:
                file.write(text)

        (...)

    return mainCorpus
```

Listagem 3.5: Função `corpusCombinatorMain`, responsável pela combinação dos vários *corpora* principais e formação do *corpus* 1. Desenvolvido neste trabalho.

### 3.2.5 Menu de Processamento do *Corpus*

O menu de processamento do *corpus* é uma forma simples de centralizar todo o processo de limpeza invés de percorrer vários ficheiros responsáveis pela realização de pequenas etapas associados ao mesmo. O menu de processamento do *corpus* é um simples programa de consola que permite duas ações:

1. Limpar os *corpora* individuais de notícias e relatórios dentro de um intervalo de anos específico.

2. Criar o *corpus* principal a partir dos *corpora* individuais limpos e processados para um determinado intervalo de tempo.

Ao mesmo tempo que expedita o processo de limpeza, o menu gere a chamada e o momento de chamada de funções, garantido também que cada uma recebe sempre a informação correta no formato correto. O aspeto da consola quando se corre o programa encontra-se na figura 3.3. Em primeiro lugar, têm-se as ações possíveis de serem realizadas acompanhadas de informações úteis à utilização do menu.

```
What would you like to do?
1 - Clean the individual corpora of the news and articles
2 - Combine individual processed news and articles into a main clean corpus
```

Figura 3.3: Aspeto inicial do menu de processamento do *corpus*. As ações possíveis de serem realizadas são apresentadas em primeiro lugar acompanhadas de avisos e instruções para o uso correto do menu.

Após a escolha da ação a realizar, surge outro campo de preenchimento onde se especifica qual é o intervalo de tempo a processar. Mais uma vez, o campo é acompanhado de instruções sobre como indicar esse intervalo. A flexibilidade do menu permite não só limpar ou combinar os *corpora* de um só ano, como realizar as mesmas ações para vários anos de uma só vez. Escolhida a ação e o intervalo para a realizar, o utilizador recebe um conjunto de mensagens indicativas do estado atual do processo. Por exemplo, caso se escolhesse combinar os *corpora* individuais do intervalo entre 2000 a 2004, o aspeto do menu seria como o da figura 3.4.

```
What would you like to do?
1 - Clean the individual corpora of the news and articles
2 - Combine individual processed news and articles into a main clean corpus
2

Input an year from 1987 to 2020 (an interval of the type ####-####, ex. 2000-2008, will
 perform the action in bulk): 2000-2004

Corpora from 2000 combined.
Corpora from 2001 combined.
Corpora from 2002 combined.
Corpora from 2003 combined.
Corpora from 2004 combined.
```

Figura 3.4: Aspeto do menu de processamento do *corpus* quando se escolhe combinar os vários *corpora* entre 2000 a 2004. Cada vez que os *corpora* são combinados com sucesso, é enviada uma mensagem a indicá-lo.

Por fim, e depois da ação terminar, surge o um último menu onde é fornecida a escolha entre terminar o programa ou continuar para o menu inicial.

## 3.3 Produção de Resultados - *Corpus* 1

Para todas as etapas dentro da produção de resultados para o *corpus* 1 a organização de ficheiros é igual à dos ficheiros para o processamento do *corpus*. Existe sempre um ficheiro responsável por guardar todas as funções associadas à criação de um determinado conjunto de resultados e um outro responsável por agilizar o processo de acesso a essas funções. Este último traduz-se num menu onde o utilizador pode facilmente indicar e

produzir os resultados desejados. Caso seja necessário saltar a utilização dos menus, os ficheiros de funções encontram-se documentados com todas as funções devidamente identificadas.

### 3.3.1 Gráficos Globais

Entenda-se por gráfico global um gráfico onde estão ilustradas todas as palavras de um modelo após uma redução de dimensão para 3D ou 2D cujo algoritmo de projecção foi descrito em 2.1.5.

Os gráficos globais são produzidos tendo em conta um intervalo de tempo e os sub-intervalos que o compõem. Como para cada intervalo de anos é gerado um modelo, os gráficos globais são todos baseados no mesmo modelo e, consequentemente, são uma medida qualitativa da evolução do mesmo. Daqui, conclui-se naturalmente que uma característica básica destes gráficos seja a partilha de limites entre os mesmos para uma maior facilidade na comparação entre eles.

Contudo, uma outra vantagem da produção dos gráficos globais é a produção de dados adicionais que permitem a análise de outras características, sendo guardados para uso futuro. Em suma, os dados produzidos para cada sub-intervalo são:

- Um modelo de palavras treinado com base no *corpus* 1 associado ao sub-intervalo em questão com todas as palavras e as suas coordenadas no espaço de elevada dimensão.
- Uma matriz global construída a partir do modelo com todas a palavras e a suas coordenadas no espaço de elevada dimensão.
- Uma ou mais matrizes com todas as palavras e as suas posições no espaço de dimensões reduzidas, podendo ser no máximo quatro matrizes.
- Um ou mais ficheiros csv com a mesma informação que as matrizes de dimensão reduzida, podendo ser no máximo quatro.
- Uma ou mais figuras com os gráficos globais.

Os gráficos globais, tais como todos os gráficos que serão referidos no futuro podem ser de vários tipos em termos de dimensão e fórmulas utilizadas no cálculo da redução da dimensão. Primeiro, os gráficos podem ser 2D ou 3D, segundo e com o intuito de estudar como é que a normalização afeta os vetores das palavras, o algoritmo de t-SNE pode utilizar nos seus cálculos de redução a similaridade de cosseno ou o produto interno. Assim, a criação dos gráficos globais é realizada em três etapas: criação dos ficheiros de dados, carregamentos dos ficheiros e criação dos gráficos.

Na etapa inicial, são gerados os ficheiros de dados para todas as variações possíveis através da função `dataCreator`, cabeçalho na listagem 3.6. Apesar do processo de geração de dados ser moroso, esta abordagem garante que todos os ficheiros que possam vir a ser necessários no futuro são criados de imediato, eliminando a necessidade de passar pelo processo de criação outra vez.

```
def dataCreator(beg, end, checkP, saveFolder):
    # Creates and trains a model for a given year interval, saving the model
    state in a dataframe at user-defined checkpoints. The calculations are
    based on the cosine similarity and dot product
    # beg        - first year of the year interval
    # end        - last year of the year interval
    # checkP     - value for the number of years to pass before a checkpoint
    is reached and the model is saved
    # saveFolder - folder to save data
    # return     - none
    (...)
```

Listagem 3.6: Função `dataCreator`, responsável pela criação de ficheiros de dados necessários à criação dos gráficos globais para o *corpus* 1. Desenvolvido neste trabalho.

Fornecendo o intervalo de anos à função e a posição dos locais de gravação do modelo, isto é, a informação em como dividir o intervalo principal, a função cria uma lista de anos. Através de um ciclo, a lista é percorrida e são gerados todos os nomes necessários à gravação dos dados como o nome da pasta onde guardar os ficheiros, `saveFolder`, e o nome do modelo, `modelName`. É também nesta etapa e dentro deste ciclo que o *corpus* 1 é criado e atualizado. O modelo em si é gerado de raiz apenas na primeira iteração, pois em todas as outras é sempre o modelo do estágio anterior que é treinado com o novo *corpus*. A listagem 3.7 mostra como o processo de criar e treinar modelos funciona.

```
(...)
# Check if trained model already exists
if (os.path.exists(os.path.join(saveFolderName, modelName + ".w2v"))):
    model = w2v.Word2Vec.load(os.path.join(saveFolderName, modelName + ".w2v")
    )
else:
    # If no model exists, create one

    # Check if it is the first pass, if yes then create the model
    if (yearList[index] == beg):
        # Create Model
        model = ModelTraining.createModel(corpusPath, saveFolderName,
    modelName)

    else:
        # Load the previous model and train it again
```

```
15          model = ModelTraining.createModel(corpusPath, saveFolderName,
    modelName, model)
16 (...)
```

Listagem 3.7: Ciclo responsável pela criação e treino de modelos dentro da função `dataCreator`. Desenvolvido neste trabalho.

Depois do modelo criado, este é utilizado na geração das matrizes e *dataframes* necessárias à criação das várias variações dos gráficos globais. Em primeiro lugar, é criada uma matriz para guardar as palavras e suas respetivas posições no espaço de menor dimensão, de seguida e através da matriz, cria-se uma *dataframe* e guarda-se a mesma no formato csv. A listagem 3.8 exemplifica a criação e gravação dos ficheiros para o caso em que se reduz a dimensão para 2D com base na similaridade de cosseno. O processo é análogo para as outras opções.

```
1 (...)
2 if (not os.path.exists(os.path.join(saveFolderName, "dataframeComplete2D_" +
    str(index + 1) + ".csv"))):
3             # Reduce the dimension and produce a 2D dataframe - calculations
    cosine similarity
4             matrix = DimensionReduction.reduceDimension(model, saveFolderName,
     2, "cosine")
5             dataframe2D = WordSpacePlot.createDataframe(model, matrix, 2)
6             dataframe2D.to_csv(os.path.join(saveFolderName, "
    dataframeComplete2D_" + str(index + 1) + ".csv"), index = False)
7             print("Global 2D dataframes created - cosine similarity")
8 (...)
```

Listagem 3.8: Gravação da matriz e *dataframe* necessárias à criação de um gráfico global de palavras 2D com uma redução de dimensão baseada na similaridade de cosseno. Excerto proveniente da função `dataCreator`. Desenvolvido neste trabalho.

É de notar que tanto na listagem 3.7 como na listagem 3.8 se verifica uma característica interessante e extremamente importante da função `dataCreator`. Em cada iteração em que o modelo ou outro ficheiro são criados e guardados, é sempre verificada a sua existência dentro da pasta de gravação apesar de a função ser a única capaz de produzir os mesmo. Esta insistência é uma salvaguarda para a o caso em que o processo é interrompido de forma brusca. Todos os ficheiros são criados a partir dos modelos e estes são gerados por recorrência, uma interrupção brusca implica perder a informação dos modelos anteriores. Se houver uma verificação da existência de ficheiros, a função pode saltar a sua criação e até carregar os modelos anteriores, recuperando a sua informação na eventualidade de uma paragem inesperada.

A etapa seguinte é o carregamento dos ficheiros criados e é gerida pela função `loadData`, cabeçalho na listagem 3.9. É a partir desta etapa que se faz a escolha dos ficheiros que se

pretende carregar.

```
def loadData(beg, end, checkP, metric, dimension, saveFolder):
    # Loads the dataframes previously created by dataCreator based on the
    metric chosen and dimension
    # beg        - first year of the year interval
    # end        - last year of the year interval
    # checkP     - value for the number of years to pass before a checkpoint
    is reached and the model is saved
    # metric     - metric to consider when loading dataframes ("cosine" -
    loads dataframes created based on the cosine similarity, "dot" - loads
    dataframes created based on the dot product, "both" - loads all dataframes)
    # dimension  - dimension to load (2 - loads 2D dataframes, 3 - loads 3D
    dataframes, "both" - loads all dataframes)
    # saveFolder - folder to retrieve data from
    # return     - lists with all the years to parse, chosen dataframes and
    their global limits dictionary
(...)
```

Listagem 3.9: Função `loadData`, responsável pelo carregamento dos ficheiros de dados. Desenvolvido neste trabalho.

Tal como na função `dataCreator`, o primeiro passo de `loadData` é criar uma lista com todos os anos a considerar para o intervalo de trabalho. As listas que guardarão os ficheiros carregados também são criadas neste momento juntamente com os dicionários que conterão a informação sobre os limites globais, partilhados por todos os gráficos da mesma variação. O passo seguinte é ir à pasta onde se encontram os ficheiros e carregar os ficheiros requisitados, colocando-os na lista correta. Tudo isto é feito com recurso a um ciclo onde o excerto associado ao carregamento dos ficheiros 2D baseados na similaridade de cosseno se encontra representado na listagem 3.10.

```
(...)
# Load all data
for folder in dataFolders:
    # Grab all data files
    dataFiles = os.listdir(os.path.join(saveFolder, folder))

    for file in dataFiles:
        file = file.split("_")

        (...)
        if (dimension == 2 or dimension == "both"):
            # Load dataframeComplete2D
            if(file[0] == "dataframeComplete2D" and not file[1] == "dot" and (
    metric == "cosine" or metric == "both")):
                dataComplete = pd.read_csv(os.path.join(saveFolder, folder, "_
    ".join(file)))
                dataListComplete2D.append(dataComplete)
```

```
16
17                  # Calculate the global 2D limits, the limits are shared by all
        global plots
18                  if (limitsDict2D["limMinX"] > min(dataComplete.x)):
19                      limitsDict2D["limMinX"] = min(dataComplete.x)
20
21                  if (limitsDict2D["limMaxX"] < max(dataComplete.x)):
22                      limitsDict2D["limMaxX"] = max(dataComplete.x)
23
24                  if (limitsDict2D["limMinY"] > min(dataComplete.y)):
25                      limitsDict2D["limMinY"] = min(dataComplete.y)
26
27                  if (limitsDict2D["limMaxY"] < max(dataComplete.y)):
28                      limitsDict2D["limMaxY"] = max(dataComplete.y)
29
30                  print("Global 2D dataframes loaded - cosine similarity")
31          (...)
32  (...)
```

Listagem 3.10: Pequeno excerto da função `loadData` com parte do ciclo responsável pelo carregamento de ficheiros. O código representado corresponde ao carregamento dos ficheiros 2D com base na similaridade de cosseno. Desenvolvido neste trabalho.

O último passo é preparar as listas para o retorno da função. Cada dimensão tem uma lista final associada que é a junção das listas com os dados cujos cálculos foram baseados na similaridade de cosseno e das listas de dados cujos cálculos foram baseados no produto interno. De modo a que haja a distinção entre a mudança de base nos cálculos na lista final, é acrescentado -1 antes dos ficheiros baseados no produto interno sempre que estes existam.

Após todos os dados carregados, é finalmente possível gerar os gráficos globais. Ao contrário dos processos anteriores onde a mesma função funciona para ambas as dimensões, para a criação dos gráficos existem duas funções: uma para 2D e outra para 3D. A título de brevidade, é apenas apresentada a função responsável pela produção dos gráficos globais 2D, `plotGlobal2D` cujo cabeçalho se encontra na listagem 3.11.

```
1  def plotGlobal2D(yearList, dataListComplete, limitsDict, show = True, save =
       False, saveFolder = None):
2      # Creates the global 2D plots with all the words
3      # yearList         - list with all the years in the interval to consider
4      # dataListComplete - list with the complete dataframes with all the words
       in the model for each plot
5      # limitsDictList   - dictionary with the global limits for all plots
6      # show             - wether to show plots or not
7      # save             - wether to save plots or not
8      # saveFolder       - folder to save data
9      # return           - none
```

```
(...)
```

Listagem 3.11: Função `plotGlobal2D`, responsável pela criação de gráficos globais 2D. Desenvolvido neste trabalho.

Com a inicialização de todas as variáveis de controlo como `dot` e `repIndex`, é dado início a um ciclo que percorre todas as entradas das listas de ficheiros previamente criadas. Com cada iteração é criada uma figura cujos título, subtítulo, dados e limites são determinados consoante a base dos cálculos. Na listagem 3.12 encontra-se a zona do ciclo que cria o gráfico em si.

```
(...)
fig, ax = plt.subplots(figsize=(20,20))

if (not dot):
    fig.suptitle("Metric: Cosine Similarity")
else:
    fig.suptitle("Metric: Dot Product")

# Plot the word points
ax.plot(dataComplete.x, dataComplete.y, color='crimson', marker='o', linestyle
    ='None', markersize=5, alpha=0.40)

# Defining Axes
offset = 0.0 # Change this number to adjust the plot limits in case a little
    offset is wanted for the limits
if (not dot):
    ax.set_xlim(limitsDict["limMinX"] - offset, limitsDict["limMaxX"] + offset
    )
    ax.set_ylim(limitsDict["limMinY"] - offset, limitsDict["limMaxY"] + offset
    )
else:
    ax.set_xlim(limitsDict["limMinXDot"] - offset, limitsDict["limMaxXDot"] +
    offset)
    ax.set_ylim(limitsDict["limMinYDot"] - offset, limitsDict["limMaxYDot"] +
    offset)

# Title
ax.set_title(f"{yearList[repIndex]} - {yearList[repIndex + 1]} (Number of
    words: {len(dataComplete.index)})")
(...)
```

Listagem 3.12: Zona do ciclo da função `plotGlobal2D` ao cuidado da criação dos gráficos globais 2D. Desenvolvido neste trabalho.

Caso os gráficos gerados estejam a ser criados pela primeira vez, os mesmos são sempre guardados nas pastas correspondentes e a base dos cálculos que lhes deu origem é identificada devidamente no seu nome. Para a formação dos gráficos 3D, o processo é

análogo com as devidas mudanças para acomodar a dimensão extra.

A agilização da criação dos ficheiros de dados e dos gráficos globais é feita por um menu muito semelhante ao menu de processamento do *corpus*. As ações possíveis com este menu são as seguintes:

1. Criar modelos e ficheiros de dados em 2D e 3D
2. Visualizar os dados previamente criados

A opção de visualização dos gráficos existe com o intuito de poder ver os gráficos criados a partir do menu, nomeadamente os gráficos 3D que são sempre guardados numa posição fixa que impossibilita a exploração total do gráfico. Na figura 3.5 encontra-se o aspeto inicial do menu.

```
What would you like to do?
1 - Create models and 2D or 3D dataframes in a given interval, saving them at given checkpoints and plot them (bulk creation
 available)
2 - Load and visualise previously created data

Warning:
- Creation actions (either singular or bulk) will not show the plots but will save them in the correct folders
- Never add empty space before or after commas or "-"
```

Figura 3.5: Aspeto inicial do menu de criação dos gráficos globais. Tal como para o menu de processamento do *corpus*, as ações possíveis de serem realizadas vêm acompanhadas de avisos e instruções.

Independentemente da escolha realizada, os campos de preenchimento seguintes são muito semelhantes entre si, pois o objectivo dos mesmo é perceber que dados se pretendem gerar ou carregar. exceto a opção de carregamento, a outra permite criar vários gráficos a partir de diversos parâmetros. Na figura 3.6, encontra-se o aspeto do menu após se escolher gerar dados para o período entre 2000 e 2004, criando modelos a cada 2 anos. Em cada etapa são enviadas mensagens para manter o utilizador em constante atualização. Terminada a ação, é sempre apresentado um menu de término.

```
Input if data created must be 2D, 3D or both (2 for 2D, 3 for 3D or word "both"): 2

Input if data created is based on cosine similarity, dot product or both (cosine for cosine similarity, dot for dot product
or word "both"): cosine

Input an interval from 1987 to 2020 in the form ####-#### (an interval of the type ####-#### followed by a comma and another
 interval, ex. 2000-2008,2010-2020, will perform the action in bulk over the multiple intervals): 2000-2004

Input how many years must go by until the model reaches a checkpoint and saves (must be a number smaller or equal to 10. An
input of the form #,#, no space before or after commas, will perform the action in bulk over the multiple checkpoints): 2


Creating data - 2000 to 2004 in steps of 2

Model for 2000 - 2002 created


Model for 2002 - 2004 created


Loading data...
Global 2D dataframes loaded - cosine similarity
Global 2D dataframes loaded - cosine similarity
Loading complete
```

Figura 3.6: Aspeto do menu após a escolha da opção para criar gráficos globais entre 2000 a 2004 de 2 em 2 anos. Várias mensagens são enviadas indicando o estado da ação.

### 3.3.2 Gráficos da Área Local

Um gráfico da área local é um gráfico que se foca na zona que rodeia uma palavra do espaço com o objetivo de estudar a evolução da densidade das palavras nessa área. Para o *corpus* 1, todos os gráficos locais são quadrados e construídos de modo a que a palavra central se situe no meio do quadrado. A área a estudar é variável, sendo determinada pela distância escolhida entre a palavra e a aresta do quadrado. Ao contrário dos gráficos globais que para apenas um intervalo principal são criados vários gráficos e gravados em separado, os gráficos locais são guardados numa só figura. A criação dos gráficos locais implica também a criação dos seguintes ficheiros:

- Um ou mais ficheiros csv com as palavras presentes na área a analisar e as suas coordenadas, sendo tantos quanto o número de divisões presentes no intervalo de anos a considerar ou mais se ambas as formas distintas de cálculos de redução de dimensão forem escolhidas.
- Figuras e ficheiros csv com informação geral sobre o número de palavras total do modelo, o número de palavras na área local, a palavra mais distante da palavra central, a sua distância e a distância à palavra anteriormente mais distante.
- Uma ou mais figuras com a série de gráficos locais.

O processo de criação é semelhante ao processo de criação dos gráficos globais. Primeiro são gerados os ficheiros necessários para todas as versões que se possam desejar criar no futuro. A função responsável por esse trabalho é a `squareDataCreator` cujo cabeçalho se encontra na listagem 3.13.

```
1 def squareDataCreator(center, halfEdge, yearList, dataFolder, saveFolder):
2     # Creates user-defined square plots centered in a given word for a given
      interval with saves at given checkpoints. The calculations are based on the
       cosine similarity and dot product
3     # center     - central word
4     # halfEdge   - distance from word to side of the square
5     # yearList   - list with all the years in the interval to consider
6     # dataFolder - folder to retrieve data from
7     # saveFolder - folder to save data
8     # return     - None
9 (...)
```

Listagem 3.13: Função `squareDataCreator`, responsável pela criação de ficheiros de dados necessários à criação dos gráficos locais para o *corpus* 1. Desenvolvido neste trabalho.

Os dados para os gráficos locais são gerados através dos dados anteriormente criados para os gráficos globais. Deste modo e antes de qualquer cálculo, a função carrega todos os ficheiros globais de que necessita. Após a inicialização das variáveis de controlo, a lista com os ficheiros carregados é percorrida. À medida que se avança no ciclo, é calculada a

área em redor da palavra central tendo em conta a sua distância à aresta do quadrado e uma *dataframe* com a informação sobre as palavras presentes na mesma é criada. A listagem 3.14 ilustra como a *dataframe* para um gráfico 2D é criada e os limites dos gráficos determinados.

```
(...)
# Load data
_, dataListComplete2D, dataListComplete3D, _, _ = loadData(yearList[0],
    yearList[-1], yearList[1] - yearList[0], "both", "both", dataFolder)
(...)

# Square Plots - 2D
for data in dataListComplete2D:
   (...)

    # Set the indexes of the dataframe to be the words
    data.set_index("Word", inplace = True, drop = True)

    # Calculate the 2D square plots limits
    limMinX = data.at[center, "x"] - halfEdge
    limMaxX = data.at[center, "x"] + halfEdge
    limMinY = data.at[center, "y"] - halfEdge
    limMaxY = data.at[center, "y"] + halfEdge

    # Create the square plot dataframe
    squareData = data[(limMinX <= data["x"]) & (data["x"] <= limMaxX) & (
    limMinY <= data["y"]) & (data["y"] <= limMaxY)]
(...)
```

Listagem 3.14: Parte da função `squareDataCreator` responsável pela criação dos ficheiros de dados com as palavras presentes na área de estudo. Desenvolvido neste trabalho.

A partir dos dados criados e do seu carregamento, que é realizado por uma função muito semelhante à função de carregamento dos ficheiros para os gráficos globais, é chamada a função `squareDataAnalyser` que realiza uma pequena análise dos dados. O seu cabeçalho pode ser visualizado na listagem 3.15.

```
def squareDataAnalyser(center, dataList, dataListComplete):
    # Analyses the square plots data for each plot in terms of the furthest
    word from central word and its distance, the previous furthest word and its
     new distance and the word count inside each plot
    # center           - central word
    # dataList         - list with all the square plots dataframes
    # dataListComplete - list with all the complete dataframes with all the
    words in the model
    # return           - lists with all the furthest words and their distances
    , previous furthest words and their new distances and word count of each
    square dataframe
```

```
(...)
```

Listagem 3.15: Função `squareDataAnalyser`, responsável pela análise dos dados gerados. Desenvolvido neste trabalho.

O objetivo da função `squareDataAnalyser` é simples: a partir dos dados globais e locais, calcular quantas palavras existem em cada modelo, quantas delas se encontram na área de estudo, qual a distância entre a palavra central e a palavra mais distante e entre a palavra central e a palavra anteriormente mais distante, tentado criar uma ideia da geometria do espaço e da evolução da mesma ao longo do tempo. Esta informação é também guardada tanto em forma de ficheiro csv como em forma de imagem.

Por fim, resta apenas fazer os gráficos locais. A função responsável pelos gráficos locais em 2D designa-se por `plotSquare2D`, listagem 3.16. Apesar de não se apresentar a sua versão para 3D, os seus códigos são iguais exceto em alguns locais para acomodar a dimensão a mais.

```
def plotSquare2D(center, yearList, dataList, limitsList, distWords,
    prevDistance, wordCountList, show = True, save = False, saveFolder = None,
    labels = True):
    # Creates the square 2D plots
    # center        - central word
    # yearList      - list with all the years in the interval to consider
    # dataList      - list with the square plots dataframes
    # limitsList    - list with the dictionaries with the square plots limits
    # distWords     - list with all the furthest words and their distances for
     each plot
    # prevDistance  - list with the distances of all previous furthest words
    # wordCountList - list with the word count for each square plot
    # show          - wether to show plots or not
    # save          - wether to save plots or not
    # saveFolder    - folder to save data
    # labels        - wether to show labels or not
    # return        - none
(...)
```

Listagem 3.16: Função `plotSquare2D`, responsável pela criação dos gráficos locais. Desenvolvido neste trabalho.

Ao contrário da função de criação dos gráficos globais, a função `plotSquare2D` não utiliza diretamente a lista com os ficheiros de dados carregados. Em vez disso, a lista é de imediato separada em duas, uma para os dados provenientes de reduções baseadas na similaridade de cosseno e outra para os dados provenientes de reduções baseadas no produto interno. Os gráficos são criados à medida que as listas são percorridas e, em cada um, são escritos os resultados obtidos da análise.

O menu associado aos gráficos locais tem o aspeto ilustrado na figura 3.7. As ações possíveis são, no fundo, iguais às ações do menu dos gráficos globais, apenas descritas de forma diferente. A opção 1, com o intuito de poupar tempo, permite realizar a criação de várias séries de gráficos para diferentes parâmetros de uma só vez, sendo todos os gráficos e ficheiros de dados guardados nas pastas respetivas.

```
What would you like to do?
1 - Create data for square plots with a given size for a given interval and a given word (bulk creation
 available)
2 - Load and visualise previously created data

Warning:
- Creation actions (either singular or bulk) will not show the plots but will save them in the correct
folders
- Never add empty space before or after commas or "-"
```

Figura 3.7: Aspeto inicial do menu de criação dos gráficos locais, sempre com as instruções próximas das opções disponíveis.

É de notar que, apesar de uns dos parâmetros fornecidos ser a distância entre a palavra central e a aresta do quadrado, a pasta criada para guardar os dados tem sempre no nome o tamanho da aresta do quadrado. Por exemplo, criando um conjunto de dados para o intervalo entre 2000 e 2004 dividido em sub-intervalos de 2 anos com a palavra central *economy* a 2 unidades de distância da aresta, a pasta onde os ficheiros são guardados designa-se por *economy_4* e o aspeto do menu é o presente na figura 3.7.

### 3.3.3 Gráficos de Densidade

Os gráficos de densidade são o último conjunto de gráficos a serem produzidos. Ao contrário dos anteriores, estes gráficos geram um mapa de cores que codificam as zonas de acordo com o número de palavras nelas presente. São os mais simples e os mais rápidos de serem criados e existe apenas uma função responsável pela sua criação. Essa função é designada por `densityPlot` e encontra-se descrita na listagem 3.17. Para além da produção de figuras com os gráficos, todos em 2D, não são gerados quaisquer ficheiros de dados adicionais, sendo obrigatório que todos os dados necessários já se encontrem criados.

```
1 def densityPlot(yearList, dataList, dataType, limitsList, metric, maxCB, step,
      show = True, save = False, saveFolder = None):
2     # Creates density plots based on the given data
3     # yearList   - list with all the years in the interval to consider
4     # dataList   - list with all the dataframes
5     # dataType   - dataframe type ("global" - complete dataframes with all
      words of the model, "square" - dataframes of the square models)
6     # limitsList - list with the dictionaries with the plots limits or just a
      dictionary with the global limits
7     # metric     - metric to consider when creating density plots ("cosine" -
      density plots based on the cosine similarity, "dot" - density plots based
      on the dot product, "both" - loads all dataframes)
```

```
Input if data created is based on cosine similarity, dot product or both (cosine for cosine similarity,
 dot for dot product or word "both"): cosine

Input if data created must be 2D, 3D or both (2 for 2D, 3 for 3D or word "both"): 2

Distance between word and edge of the square (bulk actions not available for this parameter): 2


Input an interval in the form ####-####-# (beginning year-end year-checkpoints, an interval of the type
 ####-####-# followed by a comma and another interval, ex. 2000-2008-4,2010-2020-2, will perform the ac
tion in bulk over the multiple intervals): 2000-2004-2

Word at the center of the squared plots (a list of words separeted by commas will perform the action in
 bulk over the multiple words): economy
Checking if words are present in the models...

Loading data...
Global 2D dataframes loaded - cosine similarity
Global 2D dataframes loaded - cosine similarity
Loading complete

Loading data...
Global 2D dataframes loaded - cosine similarity
Global 2D dataframes loaded - dot product
Global 3D dataframes loaded - cosine similarity
Global 3D dataframes loaded - dot product
Global 2D dataframes loaded - cosine similarity
Global 2D dataframes loaded - dot product
Global 3D dataframes loaded - cosine similarity
Global 3D dataframes loaded - dot product
Loading complete

Loading data...
Global 2D dataframes loaded - cosine similarity
Global 2D dataframes loaded - cosine similarity
Loading complete

Loading data...
Square 2D dataframes loaded - cosine similarity
Square 2D dataframes loaded - cosine similarity

End program (y/n)?
```

Figura 3.8: Aspeto do menu após a escolha da opção 1 para anos entre 2000 a 2004 com uma gravação de dados a cada 2 anos e para uma área quadrada de 16 unidades de área focada na palavra *economy*.

```
8     # maxCB      - max value for the highest tick of the color bar
9     # step       - difference between two consecutive ticks
10    # show       - wether to show plots or not
11    # save       - wether to save plots or not
12    # saveFolder - folder to save data
13    # return     - none
14 (...)
```

Listagem 3.17: Função `densityPlot`, responsável pela criação dos gráficos de densidade. Desenvolvido neste trabalho.

Tanto os gráficos de densidade globais como os gráficos de densidade locais são produzidos por `densityPlot`. O corpo da função é uma combinação das funções `plotGlobal2D`, listagem 3.11, e `plotSquare2D`, listagem 3.16 com uma substituição no tipo de gráfico produzido. Tome-se o exemplo da formação de um gráfico de densidade global com uma redução de densidade baseada na similaridade de cosseno. Os passos iniciais de determinação do título, subtítulo e limites são iguais aos da função `plotGlobal2D`, contudo, a produção do gráfico em si é realizada através da construção de um histograma bidimensional colorido

de acordo o número de palavras presentes em cada uma das unidades de área consideradas pelo histograma. A listagem 3.18 demonstra como é que esse histograma é criado.

```
(...)
for count, data in enumerate(dataList):
    (...)

    if (not dot):
        (...)

        # Histogram configuration
        hist, xedges, yedges = np.histogram2d(data.x, data.y, bins = 100,
    range = [[limitsList["limMinX"] - offset, limitsList["limMaxX"] + offset],
    [limitsList["limMinY"] - offset, limitsList["limMaxY"] + offset]])
        hist = hist.T
        (...)

    # Plot
    x, y = np.meshgrid(xedges, yedges)
    im = ax.pcolormesh(x, y, hist, cmap = colorMap, vmin = 0, vmax = maxCB)
    cbar = plt.colorbar(im, ticks = ticksLabel)
    cbar.ax.set_yticklabels([str(x) for x in ticksLabel])
(...)
```

Listagem 3.18: Ciclo associado à criação do histograma bidimensional da função `densityPlot`. Desenvolvido neste trabalho.

Com as suas típicas duas opções, o menu associado à produção de gráficos de densidade permite gerar e visualizar todos os gráficos, permitindo a criação de vários conjuntos de dados de seguida. O seu aspeto inicial encontra-se na figura 3.9.

```
What would you like to do?
1 - Create density plots for the 2D data and plot them (bulk creation available)
2 - Visualise previously created data

Warning:
- Creation actions (either singular or bulk) will not show the plots but will save them in the correct
folders
- In order to create the density plots, folders with data must already exist
- Never add empty space before or after commas or "-"
```

Figura 3.9: Aspeto inicial do menu de criação dos gráficos de densidade com as devidas instruções e avisos.

A particularidade do menu é a devolução de uma lista de dados possíveis de analisar após os parâmetros para os dados já terem sido escolhidos. Esta lista obriga a que a escolha final dos ficheiros base seja feita considerando as opções disponibilizadas, assegurando que não há uma geração de gráficos com base em ficheiros inexistentes. Por exemplo, se for escolhida a criação de gráficos locais para a palavra *economy*, a lista que surge é igual à da figura 3.10, aparecendo numerados todos os ficheiros disponíveis para análise. Com a escolha final feita, os gráficos são criados e surge o menu habitual de fecho.

```
Density plots for global plots or square plots ("global" - global plots, "square" - square plots):
square

Input if data created is based on cosine similarity, dot product or both (cosine for cosine similarity,
 dot for dot product or word "both"): cosine


For wich words should data be retrieved (a list of words of the form string,string will create density
plots in bulk, there shouldn't be any space between commas and words): economy

Checking if words are present in the models...
The following data was found:
1 - economy for models in 2000_2003_1 a square size of 4
2 - economy for models in 2000_2004_2 a square size of 4

Input the number or list of numbers from where the density plots will be based (no space between commas
 and numbers):
```

Figura 3.10: Lista presente no menu de criação de gráficos de densidade com todos os ficheiros disponíveis para a geração de gráficos focados na palavra *economy*.

## 3.4 Produção de Resultados - *Corpus* 2

A produção de resultados para o *corpus* 2 é muito semelhante ao processo para o *corpus* 1. A diferença entre os dois consiste na forma como o *corpus* é dividido e analisado. Para o *corpus* 1, o objetivo é analisar o *corpus* ao longo de determinados intervalos, estando o *corpus* analisado intimamente relacionado com o intervalo de análise. Para o *corpus* 2, o objetivo é estudar sempre o mesmo *corpus* mas fazê-lo dividindo-o em várias partes iguais. No fundo, o objetivo dos dois estudos é o mesmo, estudar como a relação entre palavras se altera à medida que se adicionam mais palavras ao espaço, a forma como o objetivo é alcançado é que difere de um *corpus* para o outro. Assim, as funções associadas a este *corpus* são muito semelhantes, em algumas áreas, iguais, às funções do *corpus* 1.

Para a criação dos gráficos globais, a mudança mais evidente em comparação com a função `dataCreator` da listagem 3.6 é a forma como o *corpus* é importado e analisado. Antes de se entrar no ciclo de criação de ficheiros de dados, o *corpus* é lido do princípio ao fim e o seu número total de palavras é contado. Daqui é calculado o número de palavras a adicionar ao *corpus* de análise tendo em conta a quantidade de partes em que se pretende dividir o *corpus* 2. Entrando dentro do ciclo, o *corpus* para cada divisão invés de ser todos os *corpora* principais de um determinado intervalo de anos, é o *corpus* anterior mais o número de palavras acrescentar. Repare-se que o número de palavras a acrescentar não consiste no número de palavras diferentes a acrescentar mas sim no número de palavras a acrescentar que sucedem às palavras do *corpus* anterior. Esse processo está ilustrado na listagem 3.19.

```
1 (...)
2 # Open corpus
3 with open(corpusFile, "r") as corpus:
4     text = corpus.read()
5
6 # Read corpus
```

```
text = tuple(text.split())
totalWords = len(text)
numbWords = math.ceil(totalWords / numPlots)
(...)

# Create and train the model
for index in range(numPlots):
    (...)

    # Load the corpus path to be used
    corpusPath = text[:(numbWords * (index + 1))]
(...)
```

Listagem 3.19: Importação do *corpus* 2 dentro da função `dataCreator` do *corpus* 2. Desenvolvido neste trabalho.

No caso dos gráficos locais, é a maneira como as palavras nas redondezas da palavra central são localizadas que se altera. No final do trabalho surgiu a ideia de verificar as palavras mais próximas da palavra central considerando a sua similaridade de cosseno. Assim, foi adicionada na função de criação dos gráficos locais essa possibilidade, sendo escolhida sempre que a distância entre a palavra central e a aresta do quadrado seja menor que 1. O corpo da função responsável por esse cálculo situa-se na listagem 3.20. Neste modo de localização, a área com as palavras recolhidas não é calculada de forma a que a palavra central se situe no centro do quadrado logo o cálculo de limites não é realizado.

```
(...)
# Load models
modelList = loadModels(dataFolder)

for count, model in enumerate(modelList):
    # List with words within cosine similarity interval
    wordsList = []

    for word in model.wv.vocab:
       sim = model.wv.similarity(center, word)
       if (sim >= halfEdge):
           wordsList.append(word)
(...)
```

Listagem 3.20: Localização das palavras próximas da palavra central através da sua similaridade de cosseno. Proveniente da função `squareDataCreator` do *corpus* 2. Desenvolvido neste trabalho.

Tal como para o *corpus* 1, em todas as etapas de criação de gráficos há menus para facilitar o processo, todos muito semelhantes aos menus descritos até ao momento. Contudo, em oposição ao *corpus* 1, não existe a possibilidade criar vários conjuntos de dados a partir de vários parâmetros de uma só vez. Essa funcionalidade foi removida devido ao *corpus* 2 ter um tamanho elevado e cada conjunto de dados demorar um tempo considerável a formar.

Capítulo 4

# Resultados e Discussão

O capítulo 4 resume e explica todos os resultados obtidos com o código descrito em 3. Este capítulo está dividido em duas partes: a primeira concentra todos os resultados associados ao *corpus* 1, enquanto que a segunda explica os resultados do *corpus* 2. Independente do *corpus* a trabalhar, devido ao grande tamanho das imagens, quase todas se encontram nos apêndices A, para o *corpus* 1, e B, para o *corpus* 2.

## 4.1 *Corpus* 1

O estudo do *corpus* 1 é feito considerando três períodos de tempo diferentes: um entre 1987 e 2000, outro entre 2000 e 2010 e o último entre 2010 e 2020. O intervalo principal e para o qual os resultados se focam maioritariamente é o intervalo entre 2000 e 2010 visto permitir comparar todas as observações com um período futuro e um período passado. Todos os intervalos considerados foram divididos em sub-intervalos de 2 anos.

### 4.1.1 Área de Estudo Igual, Palavras Diferentes

Sendo o universo das palavras tão extenso, a tarefa de o estudar é mais fácil se se começar por entender o seu comportamento local. Entendendo o comportamento local é depois possível fazer uma generalização para o espaço global. Desta forma, os primeiros resultados obtidos são os gráficos locais.

Os gráficos locais não são nada mais que uma área ou volume centrados numa palavra à escolha. A escolha das palavras centrais teve em conta o teor económico e financeiro do *corpus*, por isso as três palavras usadas como centro das áreas a estudar foram *risk*, *bank* e *financial*. Em relação à área estudada, cujos cálculos foram feitos com base na distância euclidiana, considerou-se um quadrado de 64 unidades de área (64 u.a.) e 256 unidades

Tabela 4.1: Densidade de palavras para uma área centrada em *risk* (*Corpus* 1; Área: 64 u.a.; Intervalo: 2000-2010; Redução: Similaridade de Cosseno).

| **Área de 64 u.a. - Foco em *risk* (Redução com Similaridade de Cosseno)** | | | | | |
|---|---|---|---|---|---|
| **Dados** | **2000 - 2002** | **2002 - 2004** | **2004 - 2006** | **2006 - 2008** | **2008 - 2010** |
| **Nº Total de Palavras** | 1970 | 2559 | 3053 | 3677 | 4226 |
| **Nº Palavras Dentro da Área** | 26 | 3 | 19 | 27 | 41 |
| **Densidade (palavras / u.a.)** | 0,41 | 0,05 | 0,30 | 0,42 | 0,64 |

de área (256 u.a.). Visto o espaço de palavras não ter unidades atribuídas e ainda não se conhecer muito bem o significado de distância dentro do espaço, as unidades de medidas serão sempre muito gerais.

Os primeiros resultados foram obtidos para uma área de 64 u.a. centrada na palavra *risk* e encontram-se na figura A.1. O intervalo considerado foi o de 2000 a 2010 com gráficos a cada 2 anos. A redução de dimensão para 2D foi feita com base na similaridade de cosseno. Antes de quaisquer resultados terem sido obtidos, o que se esperava era encontrar um aumento de densidade à medida que se avançava no tempo pois o avanço no eixo temporal implicava um aumento de palavras dentro do modelo. Contudo, os resultados obtidos foram surpreendentes, o que provocou uma mudança na teoria adotada.

De imediato se percebe que a densidade não teve um aumento como esperado. De 2000-2002 para 2002-2004 a densidade diminuiu bruscamente e só depois começou a aumentar. Contudo, o aumento observado não foi muito substancial. Na tabela 4.1 são apresentados os valores de densidade para os gráficos criados.

Para além disso, a localização da palavra central também está sempre em constante mudança, havendo momentos em que as suas coordenadas quase que se alteram para o seu simétrico. O mesmo ocorre para as palavras presentes dentro da área a considerar. À exceção da palavra *risk*, dos cinco gráficos criados, só o de 2004-2006 é que partilha palavras com o de 2006-2008 (*liquid* e *preference*) e com o de 2008-2010 (*risks* e *upside*).

Com o objetivo de verificar se o mesmo ocorre para outras palavras, aplica-se o mesmo processo às palavras *bank* e *financial*. Os seus resultados encontram-se nas figuras A.2 e A.3, respetivamente. Como o intervalo de tempo é igual para todas as palavras, os modelos utilizados são os mesmos.

Tabela 4.2: Número de palavras totais e locais para os resultados obtidos para as palavras *risk*, *bank* e *financial* (*Corpus* 1; 64 u.a.; 2000-2010; Similaridade de Cosseno).

| **Número de Palavras em 64 u.a. (Redução com Similaridade de Cosseno)** | | | | | |
|---|---|---|---|---|---|
| **Dados** | **2000 - 2002** | **2002 - 2004** | **2004 - 2006** | **2006 - 2008** | **2008 - 2010** |
| **Nº Total de Palavras** | 1970 | 2559 | 3053 | 3677 | 4226 |
| **Nº de Palavras: *risk*** | 26 | 3 | 19 | 27 | 41 |
| **Nº de Palavras: *bank*** | 13 | 15 | 13 | 11 | 25 |
| **Nº de Palavras: *financial*** | 20 | 5 | 13 | 26 | 52 |

Tabela 4.3: Resumo das densidades para as palavras *risk*, *bank* e *financial*(*Corpus* 1; 64 u.a.; 2000-2010; Similaridade de Cosseno). As densidades para *risk* e *financial* seguem uma tendência relativamente semelhante. As densidades de *bank*, por outro lado, adquirem valores sempre muito próximos exceto no último sub-intervalo.

| **Densidade em 64 u.a. (Redução com Similaridade de Cosseno)** | | | | | |
|---|---|---|---|---|---|
| **Palavra** | **2000 - 2002 (palavras / u.a.)** | **2002 - 2004 (palavras / u.a.)** | **2004 - 2006 (palavras / u.a.)** | **2006 - 2008 (palavras / u.a.)** | **2008 - 2010 (palavras / u.a.)** |
| ***risk*** | 0,41 | 0,05 | 0,30 | 0,42 | 0,64 |
| ***bank*** | 0,20 | 0,23 | 0,20 | 0,17 | 0,39 |
| ***financial*** | 0,31 | 0,08 | 0,20 | 0,41 | 0,81 |

Para a palavra *financial* verificam-se oscilações semelhantes às da palavra *risk*, porém, para *bank* as oscilações são mínimas exceto no último intervalo. Um resumo dos gráficos obtidos para as três palavras situa-se na tabela 4.2, enquanto que um resumo das medidas de densidade se situa na tabela 4.3. Tal como para *risk*, apenas um ou dois gráficos de *financial* partilham palavras entre si, sendo as palavras partilhadas um número muito reduzido. Visto ser a mais consistente de todas as palavras testadas, a palavra *bank*, por outro lado, consegue ter sempre uma palavra em comum em todos os gráficos para além dela (*central*).

A análise dos resultados obtidos demonstra que a hipótese da densidade aumentar com o aumento de palavras não se verifica. Apesar do número de palavras de teste ser reduzido, em nenhuma delas se demonstrou um aumento da densidade imediato e esta sofreu sempre oscilações. O único momento onde a densidade sofreu um aumento foi

Tabela 4.4: Resumo das medidas de densidade para as palavras *risk*, *bank* e *financial* (*Corpus* 1; 64 u.a.; 2000-2010; Produto Interno). As densidades observadas sofrem todas variações muito bruscas. Com o produto interno, é ainda mais fácil de verificar que a densidade deveras não aumenta com o aumento de palavras.

| **Densidade em 64 u.a. (Redução com Produto Interno)** | | | | | |
|---|---|---|---|---|---|
| **Palavra** | **2000 - 2002 (palavras / u.a.)** | **2002 - 2004 (palavras / u.a.)** | **2004 - 2006 (palavras / u.a.)** | **2006 - 2008 (palavras / u.a.)** | **2008 - 2010 (palavras / u.a.)** |
| ***risk*** | 20 | 5,5 | 1,6 | 5,0 | 9,6 |
| ***bank*** | 11 | 7,5 | 4,7 | 7,1 | 7,0 |
| ***financial*** | 13 | 3,4 | 5,2 | 5,5 | 2,1 |

no último sub-intervalo. Para testar se as oscilações da densidade eram efeito da normalização dos dados, geraram-se os resultados da figura A.4. Para este conjunto de dados mantiveram-se todos os parâmetros iguais tirando a forma como a redução de dimensão é feita onde, em vez de se utilizar a similaridade de cosseno, se utilizou o produto interno. Devido à elevada existência de palavras dentro de cada área de estudo, foram removidos todos os rótulos das palavras menos o da palavra central. Ainda que os resultados para as outras palavras não sejam demonstrados, a tabela 4.4 representa um resumo dos mesmos.

De imediato se nota que a densidade, quando comparada com os resultados da redução com a similaridade de cosseno, é muito superior. A falta de normalização nos cálculos sugere uma maior compactação das palavras, hipótese corroborada pela figura A.5. As palavras encontram-se mais perto umas das outras mesmo em momentos em que os seus significados não são muito parecidos, resultando num aumento elevado da densidade. Para melhor visualizar este efeito, criaram-se os gráficos globais para o intervalo entre 2000 e 2010. Devido ao espaço por eles ocupado, são apresentados apenas dois de cada tipo de redução na figura A.6.

Note-se primeiro que, tal como sugerido anteriormente, a área ocupada pelas palavras na redução com base no produto interno é menor do que a área ocupada na redução baseada na similaridade de cosseno. Para ambos os casos, o primeiro sub-intervalo apresenta sempre uma forma diferente das restantes, sendo a formas dos outros sub-intervalos mais consistentes entre si e semelhantes com as apresentadas para os sub-intervalos entre 2008 e 2010.

Com a obtenção de mais dados, chega-se à conclusão que o produto interno cria uma compactação de palavras tão elevada que, em alguns casos, todas as palavras do espaço se focam num só ponto ou numa vizinhança muito pequena desse ponto. Com todas as palavras num só local, torna-se impossível estudar o espaço, logo todas as reduções de dimensão feitas de agora em diante são baseadas na similaridade de cosseno.

Tabela 4.5: Resumo dos resultados obtidos para as palavras *risk*, *bank* e *financial* (*Corpus* 1; 64 u.a.; 1987-2000, 2010-2020; Similaridade de Cosseno).

| **Densidade em 64 u.a. (Redução com Similaridade de Cosseno)** | | | | |
|---|---|---|---|---|
| **Intervalos** | **Sub-Intervalos** | ***risk* (palavras / u.a.)** | ***bank* (palavras / u.a.)** | ***financial* (palavras / u.a.)** |
| **1987 - 2000** | 1987 - 1989 | 0,77 | 0,71 | 0,61 |
| | 1989 - 1991 | 0,67 | 0,48 | 0,56 |
| | 1991 - 1993 | 0,18 | 0,38 | 0,33 |
| | 1993 - 1995 | 0,34 | 0,27 | 0,31 |
| | 1995 - 1997 | 0,59 | 0,45 | 0,44 |
| | 1997 - 1999 | 0,41 | 0,50 | 0,56 |
| | 1999 - 2000 | 0,55 | 0,47 | 0,73 |
| **2010 - 2020** | 2010 - 2012 | 0,11 | 0,23 | 0,23 |
| | 2012 - 2014 | 0,25 | 0,22 | 0,19 |
| | 2014 - 2016 | 1,4 | 1,9 | 2,1 |
| | 2016 - 2018 | 4,0 | 2,5 | 2,5 |
| | 2018 - 2020 | 5,3 | 3,4 | 3,5 |

Independentemente do tipo de redução que é feita, a densidade verificada para cada palavra nunca aumenta como esperado, existindo sempre oscilações. Para se verificar se o efeito existe apenas no intervalo entre 2000 a 2010, obtiveram-se os resultados para os intervalos entre 1987 a 2000 e 2010 a 2020. O resumo dos mesmos encontra-se na tabela 4.5.

Tanto para o intervalo 1987 a 2000 como para o intervalo 2010 a 2020 se observa de novo as oscilações na densidade. Contudo, repare-se nas entradas da tabela 4.5 de 2014 para a frente. A densidade aumenta de uma forma brusca, passando de valores menores que 1,0 palavras / u.a. para valores superiores a 5,0 palavras / u.a.. Isto pode estar relacionado com o aumento substancial de *corpora* individuais disponíveis para análise de 2015 em diante.

### 4.1.2 Área de Estudo Diferentes, Mesma Palavra

Após o estudo das várias palavras para uma área sempre do mesmo tamanho, faz sentido verificar o que ocorre quando se mantém a palavra constante e se aumenta a área de estudo. Para tal e para adicionar aos resultados já obtidos, geraram-se os dados para

Tabela 4.6: Densidade de palavras para uma área centrada em *risk* (*Corpus* 1; 256 u.a.; 2000-2010; Similaridade de Cosseno).

| **Área de 256 u.a. - Foco em *risk* (Redução com Similaridade de Cosseno)** | | | | | |
|---|---|---|---|---|---|
| **Dados** | **2000 - 2002** | **2002 - 2004** | **2004 - 2006** | **2006 - 2008** | **2008 - 2010** |
| **Nº Total de Palavras** | 1970 | 2559 | 3053 | 3677 | 4226 |
| **Nº Palavras Dentro da Área** | 109 | 54 | 47 | 111 | 166 |
| **Densidade (palavras / u.a.)** | 0,426 | 0,21 | 0,18 | 0,434 | 0,648 |

uma área de 256 u.a. focada na palavra *risk* dentro do intervalo 2000 a 2010 dividido em sub-intervalos de 2 anos. Os resultados encontram-se tanto na figura A.7 como na tabela 4.6.

Quando estes resultados são comparados com os da tabela 4.1, observa-se que, apesar da área de estudo ter quadruplicado, a densidade manteve-se semelhante à densidade anteriormente calculada na maior parte dos sub-intervalos. Considerando que a palavra não se situa em nenhuma fronteira do espaço global, é esperado que o aumento da área implique um aumento de palavras suficiente para contrabalançar o efeito anterior. Caso a palavra se encontrasse na fronteira, o aumento da área implicava incluir mais espaço vazio e, como tal, a densidade diminuiria.

É talvez mais interessante olhar para a comparação entre o espaço 2D e o espaço 3D. Para o espaço 3D, primeiro faz-se a redução de dimensão e depois procura-se as palavras dentro de um volume igual a 4096 unidades de volume (4096 u.v.) que equivale a um cubo com 16 unidades de comprimento (16 u.c.) de aresta. Os resultados para este cubo encontram-se na figura A.8 e na tabela 4.7.

Para o caso 3D, a densidade de cada sub-intervalo é sempre baixa. Além disso, a densidade não só não aumenta com o aumento de palavras como ainda sofre uma descida. O entendimento desta descida requer olhar para o espaço 3D global. Estes resultados encontram-se na figura A.9 e, mais uma vez, não são apresentados na sua totalidade visto serem de grandes dimensões.

A partir da figura A.9 verifica-se que a diminuição da densidade de palavras não é local, mas sim geral. O primeiro sub-intervalo encontra-se com uma distribuição de palavras muito mais densa do que o último sub-intervalo. É ainda possível ver a forma em

Tabela 4.7: Densidade de palavras para um volume centrado em *risk* (*Corpus* 1; 4096 u.v.; 2000-2010; Similaridade de Cosseno). Para o caso 3D, a densidade não só não aumenta, como ainda começa a diminuir.

| **Volume de 4096 u.v. - Foco em *risk* (Redução com Similaridade de Cosseno)** | | | | | |
|---|---|---|---|---|---|
| **Dados** | **2000 - 2002** | **2002 - 2004** | **2004 - 2006** | **2006 - 2008** | **2008 - 2010** |
| **Nº Total de Palavras** | 1970 | 2559 | 3053 | 3677 | 4226 |
| **Nº Palavras Dentro da Área** | 562 | 758 | 286 | 107 | 73 |
| **Densidade (palavras / u.a.)** | 0,137 | 0,185 | 0,0698 | 0,0261 | 0,018 |

Tabela 4.8: Partilha de palavras entre os gráficos locais 2D e 3D centrados na palavra *risk* (*Corpus* 1; 2000-2010; Similaridade de Cosseno). À exceção do primeiro sub-intervalo, todos os outros não possuem uma partilha completa de palavras por parte dos gráficos 2D.

| **Palavras em Comum - Gráficos Locais Focados em *risk* (Redução com Similaridade de Cosseno)** | | | | | |
|---|---|---|---|---|---|
| **Dados** | **2000 - 2002** | **2002 - 2004** | **2004 - 2006** | **2006 - 2008** | **2008 - 2010** |
| **2D** | 109 | 54 | 47 | 111 | 166 |
| **3D** | 562 | 758 | 286 | 107 | 73 |
| **Nº Palavras em Comum** | 109 | 26 | 41 | 49 | 62 |

arco visualizada na figura A.6, no entanto, ao contrário do que ocorre para 2D, o último sub-intervalo não é o que contém a maior densidade. Em relação à partilha de palavras entre os dois gráficos de dimensões diferentes, não seria estranho esperar que todas as palavras nos resultados 2D estivessem presentes no resultados 3D, pois as áreas a 2D não são nada mais que um determinado plano do cubo. Contudo, tal não ocorre como demonstrado pela tabela 4.8.

À exceção do primeiro sub-intervalo, todos os outros não possuem uma partilha completa de palavras por parte dos gráficos 2D. Perda de informação quando se faz uma redução de dimensão é habitual e faz parte do processo, porém a comparação entre 2D e 3D demonstra que até a diferença de uma dimensão entre os dois casos é o suficiente para alterar os vetores de palavras e provocar a mudança na localização geral das palavras.

Tabela 4.9: Resumo dos resultados obtidos para os gráficos locais (*Corpus* 2; 10; 512 u.v.; Similaridade de Cosseno). As palavras escolhidas foram *risk* e *bank*.

| **Densidade em 512 u.v. (Redução com Similaridade de Cosseno)** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Gráficos** | ***1*** | ***2*** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Nº Total de Palavras** | 7 513 | 10 577 | 12 731 | 14 155 | 15 256 | 16 130 | 16 927 | 17 730 | 18 405 | 19 000 |
| **Nº Palavras Locais risk** | 179 | 121 | 56 | 18 | 37 | 28 | 30 | 19 | 19 | 42 |
| **Densidade risk** | 0,350 | 0,236 | 0,11 | 0,035 | 0,072 | 0,055 | 0,059 | 0,037 | 0,037 | 0,082 |
| **Nº Palavras Locais bank** | 131 | 136 | 45 | 32 | 41 | 35 | 34 | 39 | 42 | 35 |
| **Densidade bank** | 0,256 | 0,266 | 0,088 | 0,063 | 0,080 | 0,068 | 0,066 | 0,076 | 0,082 | 0,068 |

## 4.2 *Corpus* 2

Para o estudo do *corpus* 2 não é de muita importância saber o período de tempo do *corpus*. O interesse é estudar de forma pura a densidade que melhor se revela com um *corpus* composto por um número elevado de palavras. Há falta de intervalos de tempo para dividir o *corpus*, utiliza-se o seu número total de palavras para fazer a sua separação em partes. Assim, o estudo da densidade no espaço de palavras foi feito dividindo o *corpus* em 10, 20 e 30 partes. Todos os gráficos produzidos foram em 3D com uma redução de densidade baseada na similaridade de cosseno. Para os gráficos locais foram considerados dois tipos: um de volume definido e igual a 512 u.v. e outro definido pelas palavras com uma similaridade de cosseno superior a 0.45. Para todos estes gráficos foram consideradas as palavras *risk* e *bank*.

O *corpus* tem 6 202 968 palavras após o seu processamento. Dividi-lo em 10 partes implica, em cada iteração, uma adição de 620 297 palavras ao *corpus* da iteração anterior. Note-se que não são 620 297 palavras diferentes, mas as 620 297 palavras que procedem a secção do *corpus* anterior. Estudando o *corpus* desta forma obtêm-se os resultados das figuras B.1 e B.2 e da tabela 4.9.

Com o auxílio da tabela 4.9 verifica-se, mais uma vez, que a *densidade não aumenta com o aumento de palavras*. Tal como ocorre no sub-capítulo 4.1.2, a densidade sofre uma descida à medida que se vai acrescentando palavras ao *corpus*. Após a descida de densidade, esta tem tendência a estabilizar. Para verificar se o mesmo ocorre com o *corpus* dividido em mais partes, procedeu-se à criação dos gráficos locais focados em *risk* e *bank* para 20 e 30 partes, adicionando-se 310 149 e 206 766 palavras em cada iteração, respetivamente. Os resultados para a palavra *risk* após estas divisões encontram-se nas figuras B.3 e B.4,

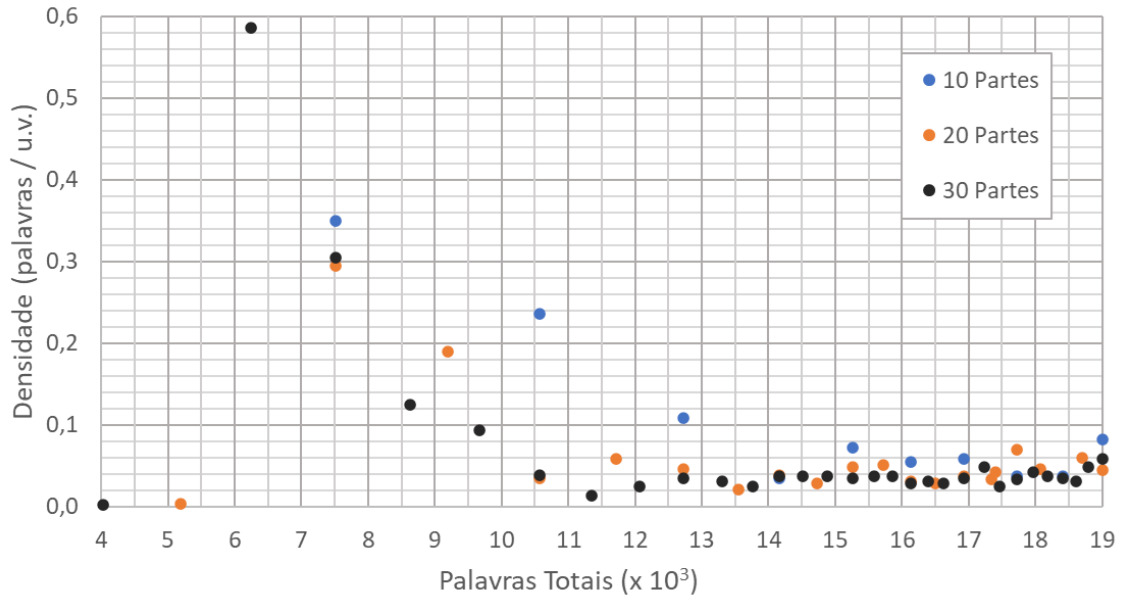


Figura 4.1: Resumo dos valores de densidade para *risk* (*Corpus* 2; Divisões: 10, 20, 30).

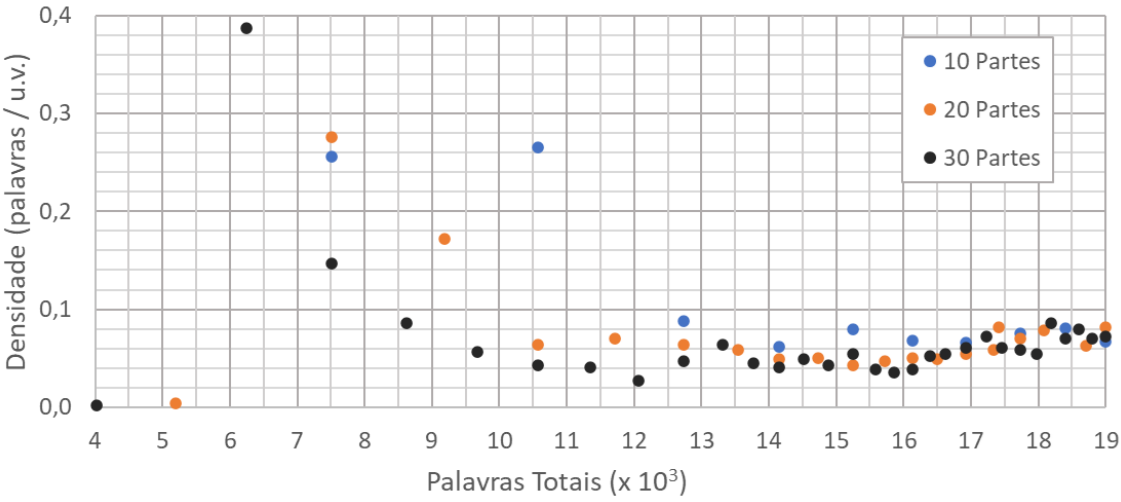


Figura 4.2: Resumo dos valores de densidade para *bank* (*Corpus* 2; Divisões: 10, 20, 30)

enquanto que os resumos das densidades para as várias divisões e várias palavras se encontram nos gráficos 4.1 e 4.2.

Ambos os gráficos demonstram que a densidade vai baixando até valores entre 0,02 a 0,04 palavras por u.v.. Uma vez neste patamar, a densidade sofre algumas oscilações mas mantém-se relativamente constante. Tanto para o *corpus* dividido em 20 partes e para o dividido em 30 partes, os dois primeiros valores da densidade estão bastante abaixo de todos os outros. Isto pode ser explicado considerando que na primeira passagem do *corpus*, quando este ainda é composto por poucas palavras, não existem palavras suficientes para criar um modelo de qualidade.

Por fim, e a título de curiosidade, foram produzidos os gráficos na figura B.5. Estes resultados, ao contrário de todos os obtidos até ao momento, foram criados tendo em conta a similaridade de cosseno entre a palavra *risk* e as outras, sendo contadas as palavras acima de uma similaridade de 0.45. Visto este parâmetro não se relacionar diretamente com uma área ou volume em específico, não foi calculada a densidade, tendo sido feita apenas uma análise qualitativa.

Através dos resultados gerados consegue-se concluir que, à medida que o número de palavras vai aumentando, o número de palavras semelhantes a *risk* aumenta também. Contudo, parece existir uma estabilização no número de palavras após ser alcançado um determinado patamar. Apesar de neste caso não se medir a densidade, a tendência para a estabilização do número de palavras semelhantes a *risk* é análoga à tendência para a estabilização da densidade dos outros resultados. Considerando que esta análise não é uma análise aprofundada talvez seja interessante, no futuro, fazer um estudo que entre por este caminho.

Capítulo 5

# Conclusão

O estudo do espaço das palavras é um estudo interessante quando se percebe que as conclusões dele retiradas podem ser os primeiros passos do estudo de assuntos com um nível de abstração muito elevado e com características semelhantes ao espaço em causa.

Aquilo que caracteriza melhor o espaço das palavras é este ser composto totalmente por elementos que não existem na natureza. Uma palavra não é nada mais que uma junção de sons ou letras que acarretam um significado que se altera com o contexto. A sua essência é tão abstrata que nem as máquinas as conseguem processar, contudo o cérebro humano é exímio a fazê-lo. Deste modo, estudar as palavras e o seu espaço requer uma abordagem que as analise da mesma forma que o cérebro humano o faz. Os melhores algoritmos de ML desenvolvidos até ao momento são as RNAs que fazem uso da representação distribuída de objetos. Neste trabalho, o algoritmo utilizado foi o modelo SG, desenvolvido por Mikolov et al. [15].

A hipótese inicial antes do estudo do espaço era que um aumento no número de palavras no espaço acarretaria um aumento de densidade das palavras. Os corpus utilizados para este efeito foram dois: um corpus composto por vários corpora organizados de forma temporal, designado por corpus 1, e outro composto por corpora suficiente para formar um corpus de, aproximadamente, $14 \times 10^6$ de palavras, designado por corpus 2. O primeiro corpus a estudar foi o corpus 1. Os gráficos locais focados em *risk*, *bank* e *financial* para o intervalo entre 2000 e 2010 provaram que a densidade não aumenta, pelo menos não de imediato. Para as três palavras se verificou que a densidade sofre uma descida imediata. Após esta descida, a densidade ou se mantém relativamente constante ou aumenta mas com muitas oscilações. Para confirmar que este efeito não era provocado pela normalização dos cálculos, criaram-se vários dados cuja redução de dimensão foi

baseada no produto interno. Os resultados demonstraram que o produto interno levava a uma maior compactação de palavras que, em alguns casos, era tão elevada que impedia o estudo correto do espaço. Assim, passou-se sempre a usar uma redução de dimensões baseadas na similaridade de cosseno. Para a comparação com outros períodos de tempo, geraram-se os resultados para o intervalo entre 1987 a 2000 e para 2010 a 2020. Para ambos se tirou as mesmas conclusões. Num primeiro momento, a densidade descia e só depois é que começava a subir, sofrendo sempre oscilações nos seus valores. Porém, é de notar que a densidade aumentava de forma significa para o intervalo entre 2010 a 2020 quando se ultrapassava o ano de 2015 e que podia ser explicado como sendo o aumento abrupto de corpora disponível para análise. De seguida, analisou-se uma área maior e comparam-se os resultados 2D com os resultados 3D. Apesar de ser esperado encontrar todas as palavras em 2D dentro dos resultados 3D o que se observou foi que a perda de informação posicional no momento em que se realizam as reduções de dimensões é suficiente para alterar a posição de várias palavras e, como tal, fazer com que palavras que em 2D se encontravam próximas da palavra central, em 3D, não estivessem dentro do volume determinado. Os gráficos 3D também demonstraram que a densidade não só não aumenta como esperado, como ainda começa a descer com o aumento de palavras.

Considerando o corpus 2, o seu estudo foi sempre feito em 3D com uma redução de dimensões baseada na similaridade de cosseno. Com a divisão do corpus em 10, 20 e 30 partes verificou-se que a densidade nunca aumenta, diminuindo nas iterações iniciais até atingir um limite onde estabiliza. Com o objetivo de satisfazer a curiosidade, foi ainda produzido um conjunto de gráficos onde as palavras selecionadas eram as que pontuavam uma similaridade de cosseno com a palavra central superior a 0,45. Neste caso, observou-se um aumento de palavras que satisfazia essa condição, porém, tal como os outros resultados, também se atingia um momento de estabilidade. No entanto, esta abordagem não foi muito estudada e ficou aberta para estudos futuros.

Generalizando e resumindo, os estudo do espaço de palavras ao longo das suas várias etapas demonstrou que o comportamento da densidade não é o que inicialmente se esperava. A densidade não aumentou de forma consistente em nenhum dos resultados obtidos, sofrendo sempre oscilações ou então estabilizando quando chegava a determinado valor. Para os resultados 3D, a densidade descia sempre até um valor dentro do intervalo entre 0,2 palavras / u.v e 0,4 palavras / u.v., mantendo-se relativamente estável daí para a frente, sugerindo que se forma uma situação de equilíbrio. A situação de equilíbrio obtida sugere que a criação de “massa” no espaço de Hilbert, leia-se a criação de mais palavras no texto, leva a uma correspondente criação de "volume".

O trabalho futuro incidirá sobre como este equilíbrio na densidade se comporta quando a dimensão do vocabulário muda no tempo, isto é, quando o espaço do vocabulário sofre, ele mesmo, um processo de inflação.

# Bibliografia


[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (ver p. ii).

[2] Priberam. *Densidade*. URL: https://dicionario.priberam.org/densidade (acedido em 2021-06-13) (ver p. 1).

[3] C. LibreTexts. *Density and its Applications*. URL: https://chem.libretexts.org/@go/page/3552 (acedido em 2021-06-13) (ver p. 1).

[4] A. Paar. *Density measurement applications*. URL: https://wiki.anton-paar.com/en/density-measurement-applications/ (acedido em 2021-06-13) (ver p. 1).

[5] E. LibreTexts. *Density of States*. URL: https://eng.libretexts.org/Bookshelves/Materials_Science/Supplemental_Modules_(Materials_Science)/Electronic_Properties/Density_of_States (acedido em 2021-06-16) (ver p. 1).

[6] A. Samitas, E. Kampouris e D. Kenourgios. «Machine learning as an early warning system to predict financial crisis». Em: *International Review of Financial Analysis* 71 (2020-10). DOI: 10.1016/j.irfa.2020.101507 (ver p. 2).

[7] J. Kang et al. «Machine learning predictive model for severe COVID-19». Em: *Infection, Genetics and Evolution* 90 (2021-06). DOI: 10.1016/j.meegid.2021.104737 (ver p. 2).

[8] A. Tiwari et al. «Using machine learning to develop a novel COVID-19 Vulnerability Index (C19VI)». Em: *Science of the Total Environment* 773 (2021-06). DOI: 10.1016/j.scitotenv.2021.145650 (ver p. 2).

[9] J. D. Aromi. «Linking words in economic discourse: Implications for macroeconomic forecasts». Em: *International Journal of Forecasting* 36 (4 2020-10), pp. 1517–1530. DOI: 10.1016/j.ijforecast.2019.12.001 (ver p. 2).

[10] D. Skinner. *Principles of Quantum Mechanics*. University of Cambridge Part II Mathematical Tripos. URL: http://www.damtp.cam.ac.uk/user/dbs26/PQM/chap2.pdf (ver pp. 4, 5).

[11] W. Greiner. *Quantum Mechanics An Introduction*. 4ª ed. Springer-Verlag Berlin Heidelberg, 2001. DOI: 10.1007/978-3-642-56826-8 (ver pp. 4, 5).

[12] R. A. Bertlmann. *Theoretical Physics T2 Quantum Mechanics*. URL: https://homepage.univie.ac.at/reinhold.bertlmann/pdfs/T2_Skript_contents_corr1.pdf (ver p. 5).

[13] IBM. *What is Natural Language Processing?* URL: https://www.ibm.com/cloud/learn/natural-language-processing (acedido em 2021-09-14) (ver p. 7).

[14] B. Kröse et al. «An introduction to neural networks». Em: *J Comput Sci* 48 (1993-01) (ver p. 7).

[15] T. Mikolov et al. «Efficient estimation of word representations in vector space». Em: International Conference on Learning Representations, ICLR, 2013-01, pp. 1–12. URL: https://arxiv.org/abs/1301.3781 (ver pp. 10, 55).

[16] T. Mikolov et al. «Distributed Representations of Words and Phrases and their Compositionality». Em: *Advances in Neural Information Processing Systems* (2013-10). URL: http://arxiv.org/abs/1310.4546 (ver pp. 11–13).

[17] L. van der Maaten e G. Hinton. «Viualizing data using t-SNE». Em: *Journal of Machine Learning Research* (2008-09) (ver p. 13).

[18] R. Albert e A.-L. Barabási. «Statistical mechanics of complex networks». Em: *Reviews of modern physics* 74.1 (2002), p. 47 (ver p. 16).

[19] J. Peng. *Analysis of Typical Elements in Various Movie Genres using TNA*. URL: https://noduslabs.com/cases/analysis-of-typical-elements-in-various-movie-genres-using-tna/ (acedido em 2021-11-17) (ver pp. 16, 18).

[20] L. Elvas. *Densidade Palavras*. URL: https://github.com/LauraElvas/DensidadePalavras (acedido em 2021-11-30) (ver pp. 20, 23).

[21] M. Bernardo. «Construction of Geometries Based on Automatic Text Interpretation». Dissertação apresentada para a obtenção do grau de Mestre em Engenharia Física à Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, 2020 (ver pp. 20, 21, 23–26).

[22] B. C. Europeu. *Relatório Anual do BCE*. URL: https://www.ecb.europa.eu/pub/annual/html/index.pt.html (acedido em 2021-09-30) (ver p. 21).

[23] B. C. Europeu. *Publicações*. URL: https://www.ecb.europa.eu/pub/html/index.pt.html (acedido em 2021-09-30) (ver p. 21).

[24] Eurocid. *Previsões económicas: Primavera, Verão, Outono e Inverno*. URL: https://eurocid.mne.gov.pt/artigos/previsoes-economicas-primavera-verao-outono-e-inverno (acedido em 2021-10-01) (ver p. 21).

[25] F. M. Internacional. *REGIONAL ECONOMIC OUTLOOK*. URL: https://www.imf.org/en/Publications/REO (acedido em 2021-09-30) (ver p. 22).

[26] Webhose. *Financial news articles*. URL: https://webhose.io/free-datasets/financial-news-articles/ (acedido em 2021-10-01) (ver p. 22).

[27] B. Data e M. L. for Finance. *News article and full details dataset*. URL: https://github.com/Finance-And-ML/News-Article-And-Full-Details-Dataset (acedido em 2021-10-01) (ver p. 22).

[28] Reuters. *Reuters-21578 Text Categorization Collection*. URL: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html (acedido em 2021-10-01) (ver p. 22).

APÊNDICE

# RESULTADOS DO *CORPUS* 1

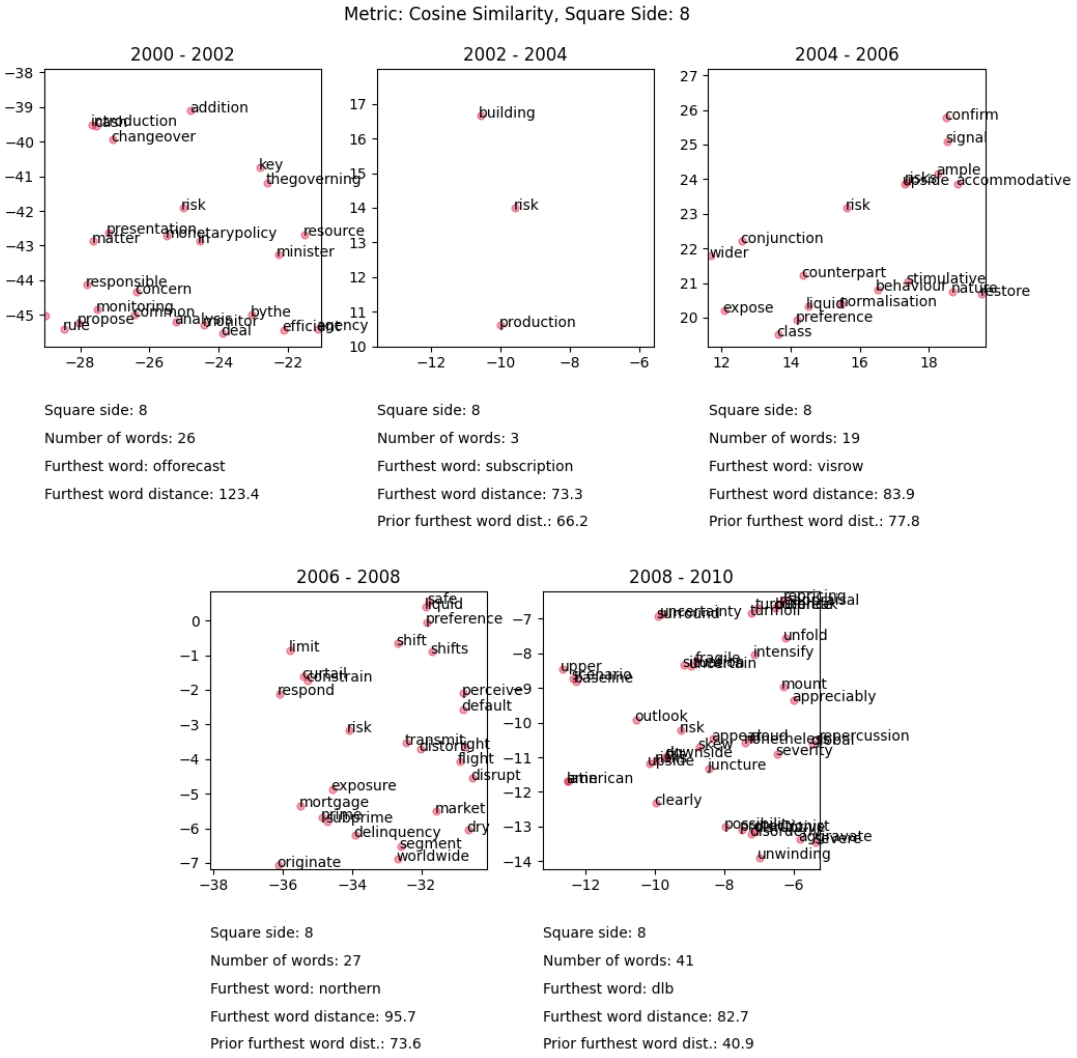


Figura A.1: Gráficos locais centrados em *risk* (*Corpus* 1; Área: 64 u.a.; Intervalo: 2000-2010; Redução: Similaridade de Cosseno). A densidade sofre algumas oscilações na passagem de um sub-intervalo para outro.

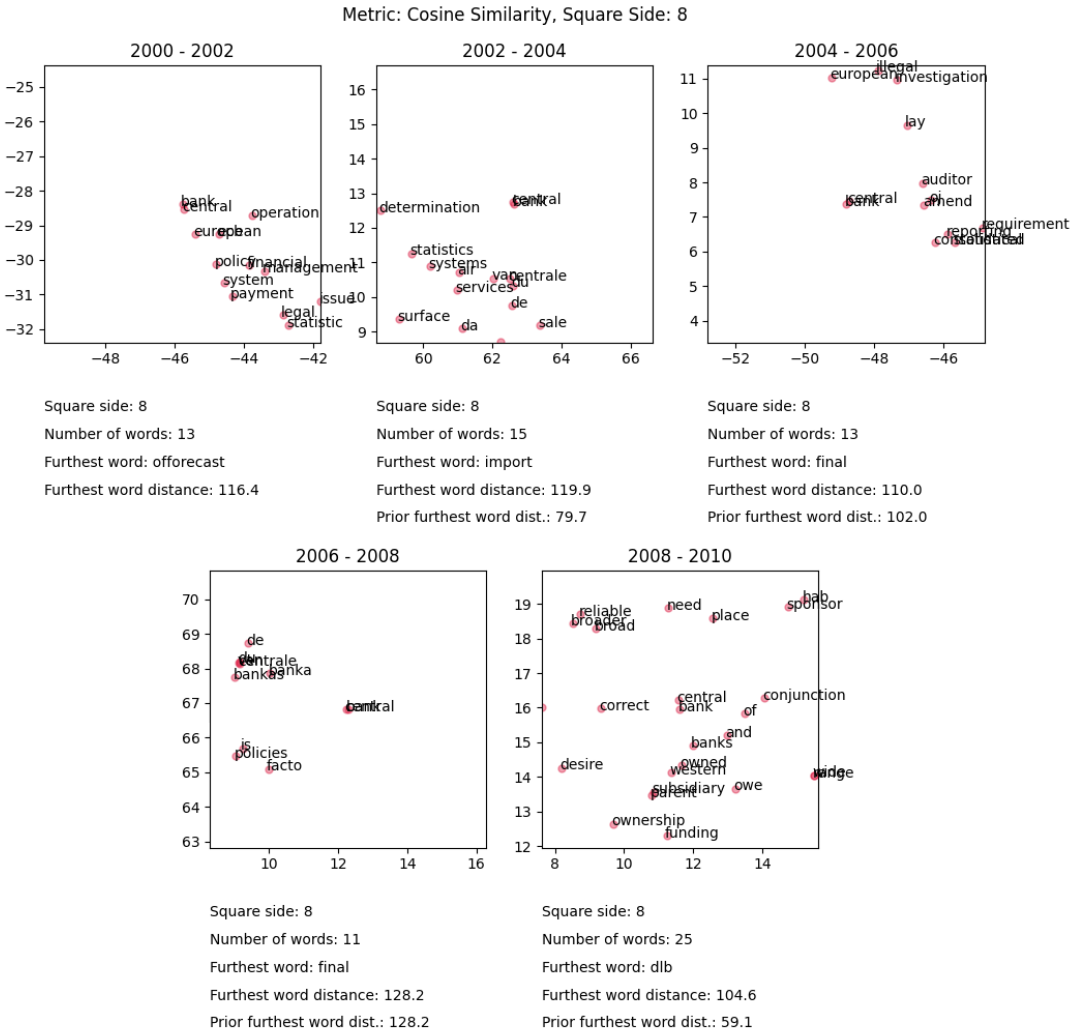


Figura A.2: Gráficos locais centrados em *bank* (*Corpus* 1; 64 u.a.; 2000-2010; Similaridade de Cosseno). A densidade observada para *bank* também parece sofrer oscilações, contudo mantém-se mais ou menos estável ao longo dos sub-intervalos.

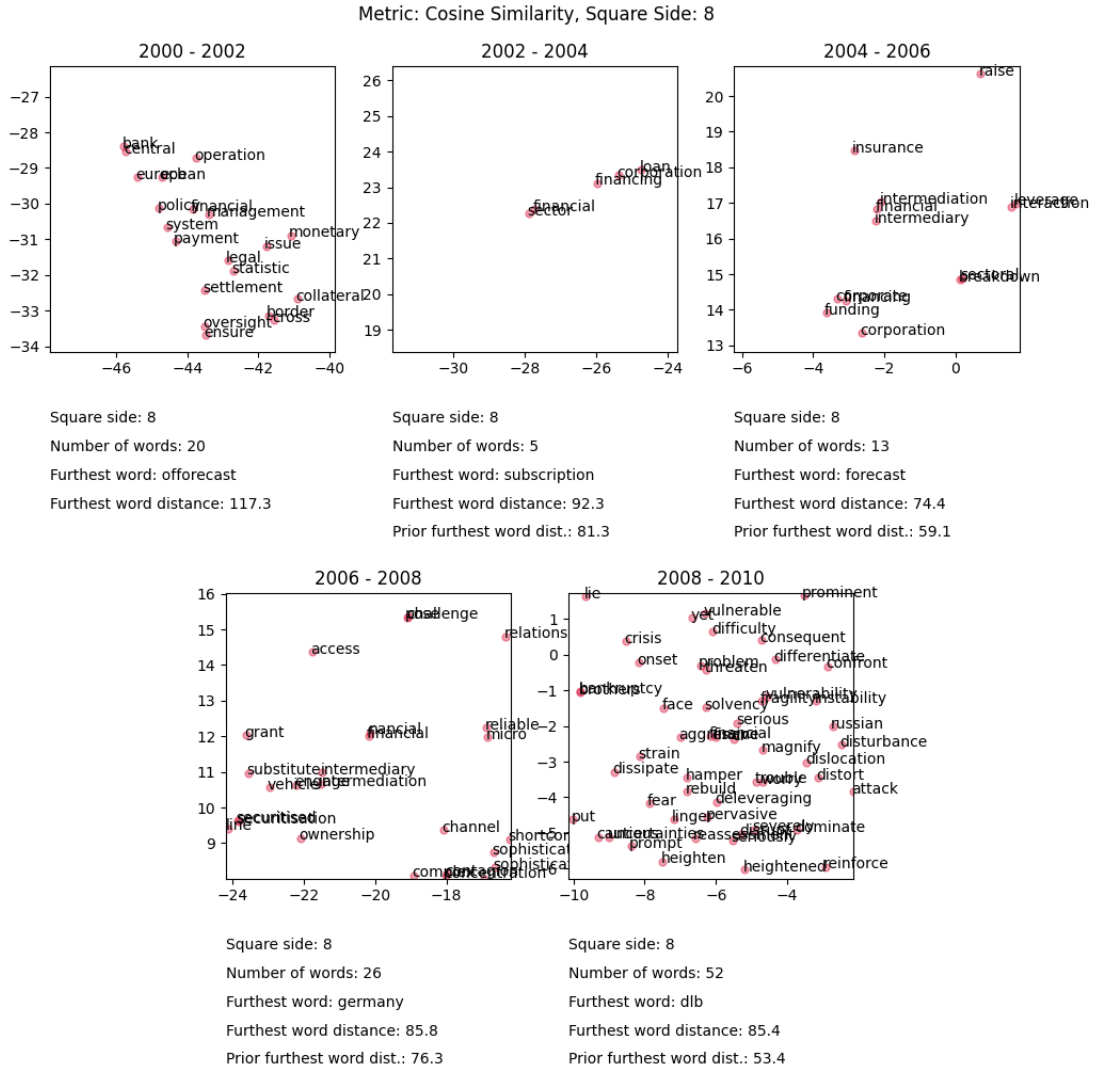


Figura A.3: Gráficos locais centrados em *financial* (*Corpus* 1; 64 u.a.; 2000-2010; Similaridade de Cosseno). Tal como para *risk*, a densidade ao longo dos sub-intervalos vai oscilando.

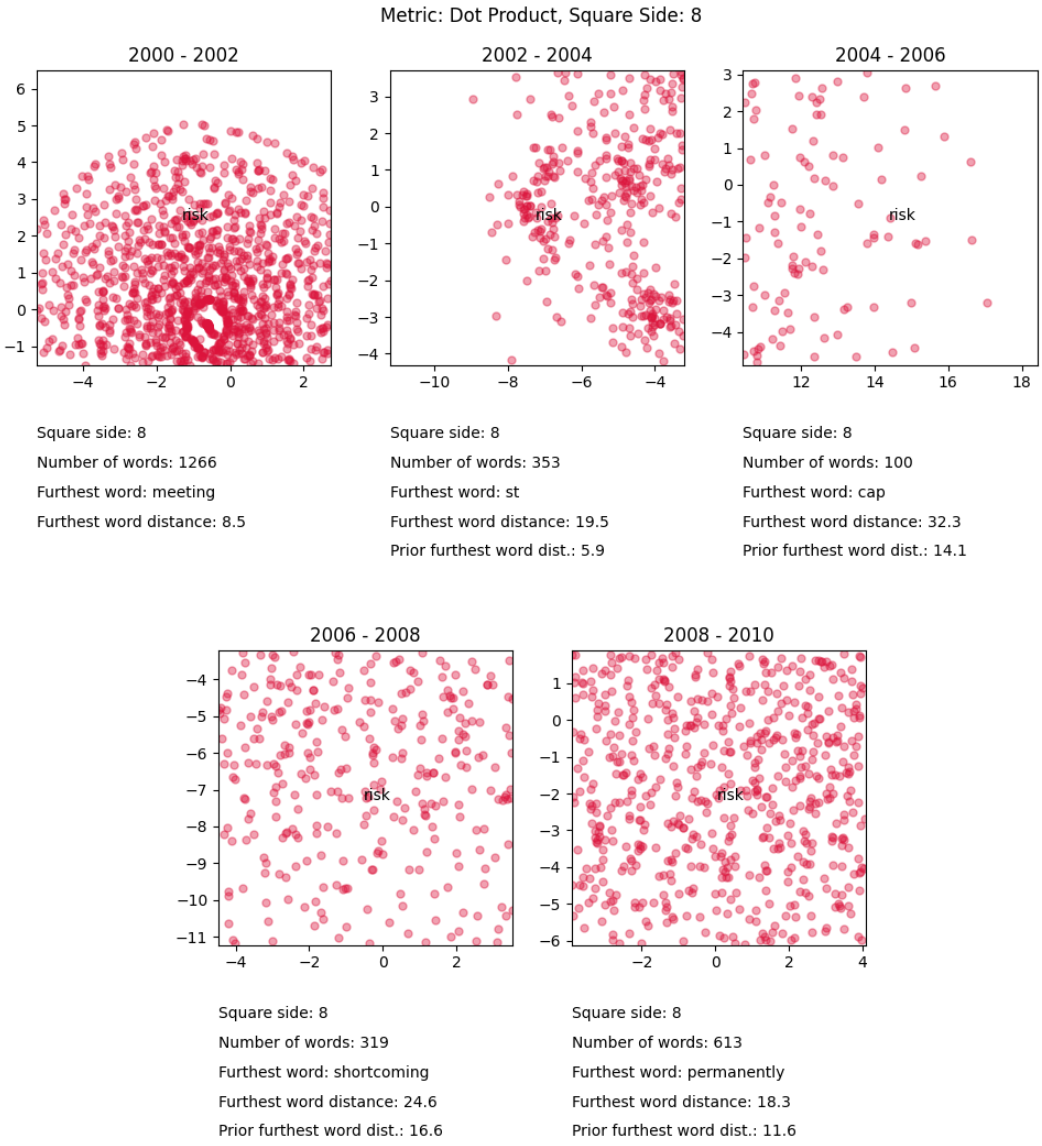


Figura A.4: Gráficos locais centrados em *risk* (*Corpus* 1; 64 u.a.; 2000-2010; Produto Interno). De sub-intervalo para sub-intervalo verifica-se uma oscilação da densidade elevada.

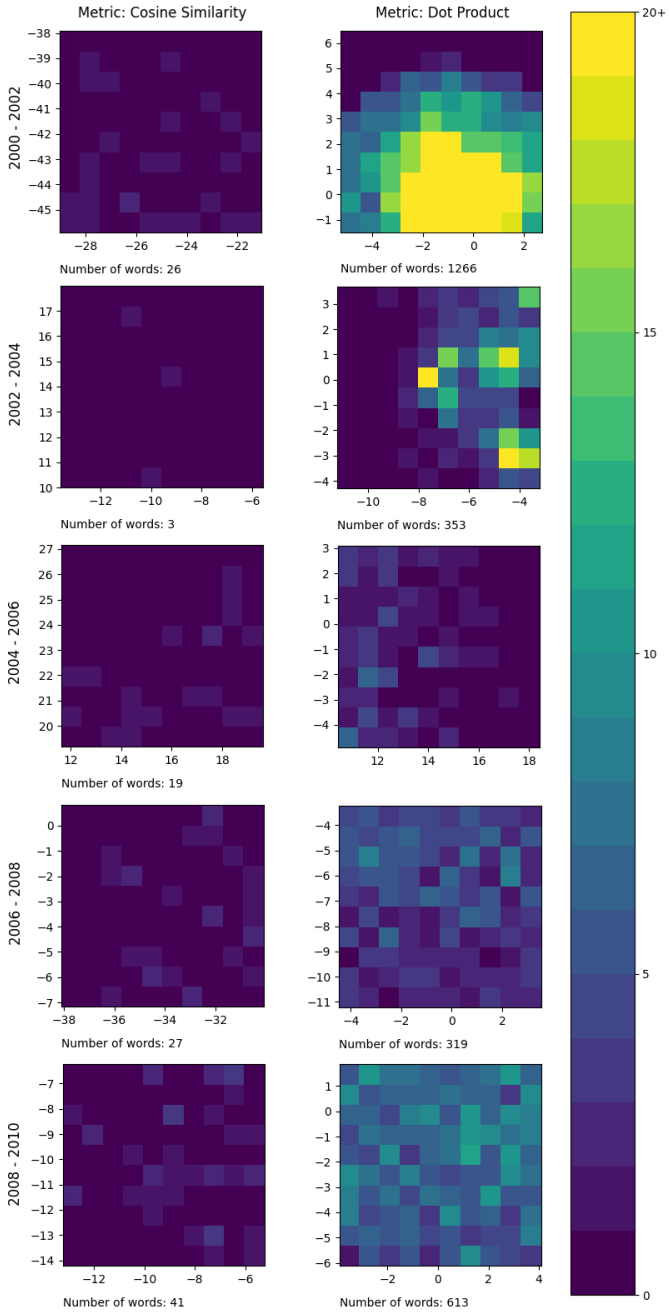


Figura A.5: Comparação da densidade dos gráficos locais obtidos através de diferentes reduções de dimensão para a palavra *risk* (2000-2010).


Figura A.6: Gráficos globais 2D dos sub-intervalos 2000-2002 e 2008-2010 do intervalo 2000-2010 com os vários tipos de redução. Dos cinco gráficos produzidos para cada tipo de redução, só os primeiros é que apresentam as formas mais diferentes. Os restantes adquirem uma forma muito mais semelhante às formas dos gráficos de baixo.

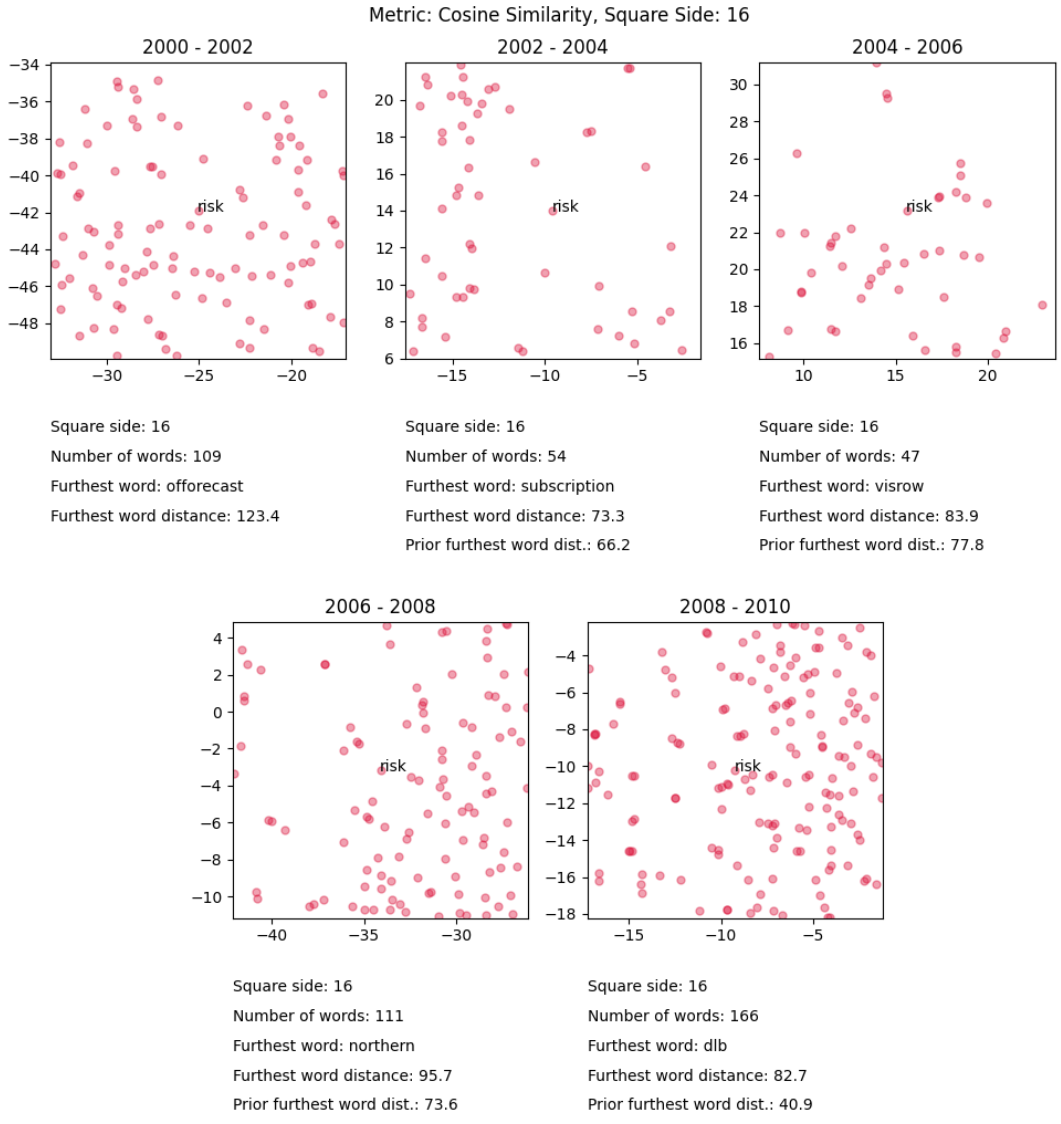


Figura A.7: Gráficos locais centrados em *risk* (*Corpus* 1; 256 u.a.; 2000-2010; Similaridade de Cosseno).


Figura A.8: Gráficos locais centrados em *risk* (*Corpus* 1; 4096 u.v.; 2000-2010; Similaridade de Cosseno). Para o caso 3D, a densidade não só não aumenta, como ainda começa a diminuir.

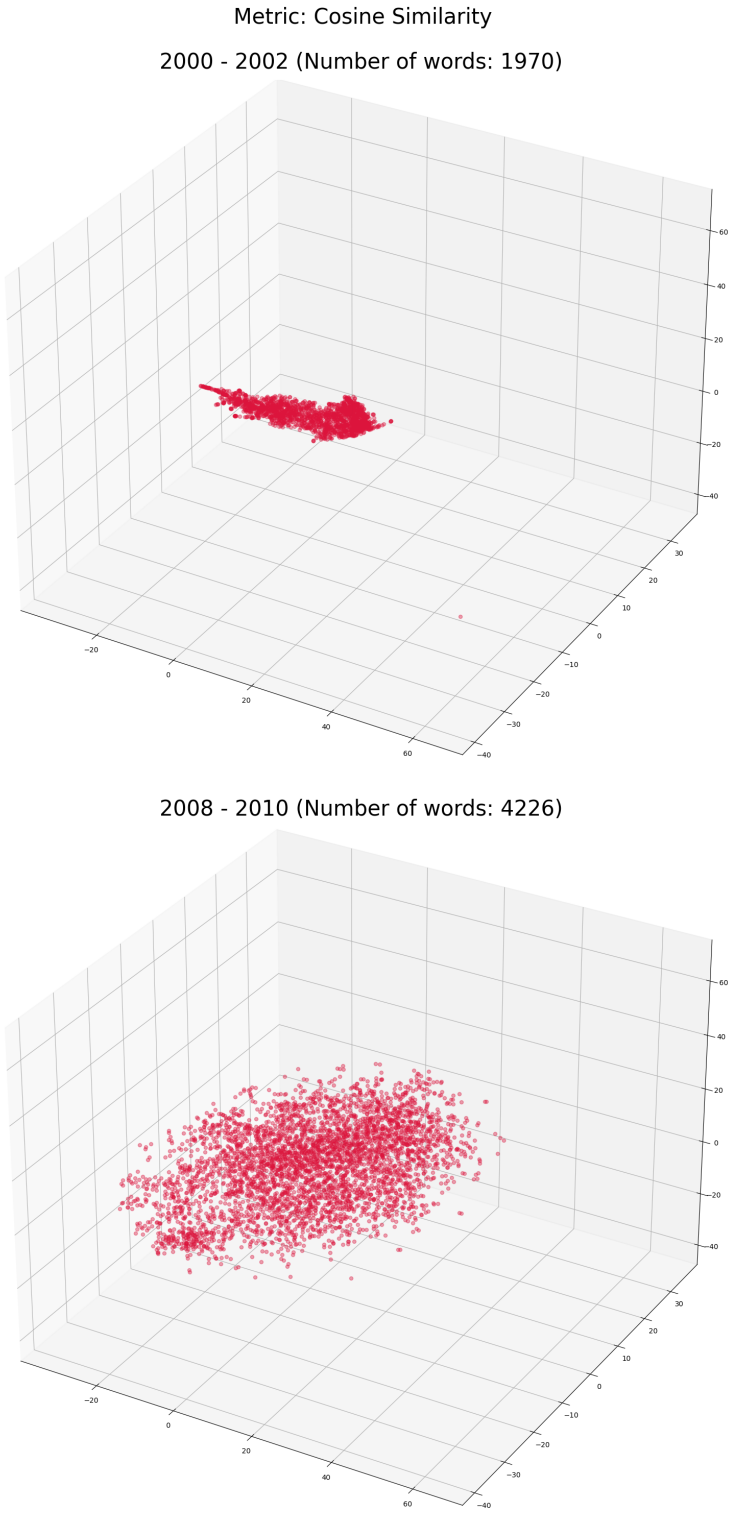


Figura A.9: Gráficos globais (*Corpus* 1; 4096 u.v.; 2000-2010; Similaridade de Cosseno).

APÊNDICE B

# RESULTADOS DO *CORPUS* 2

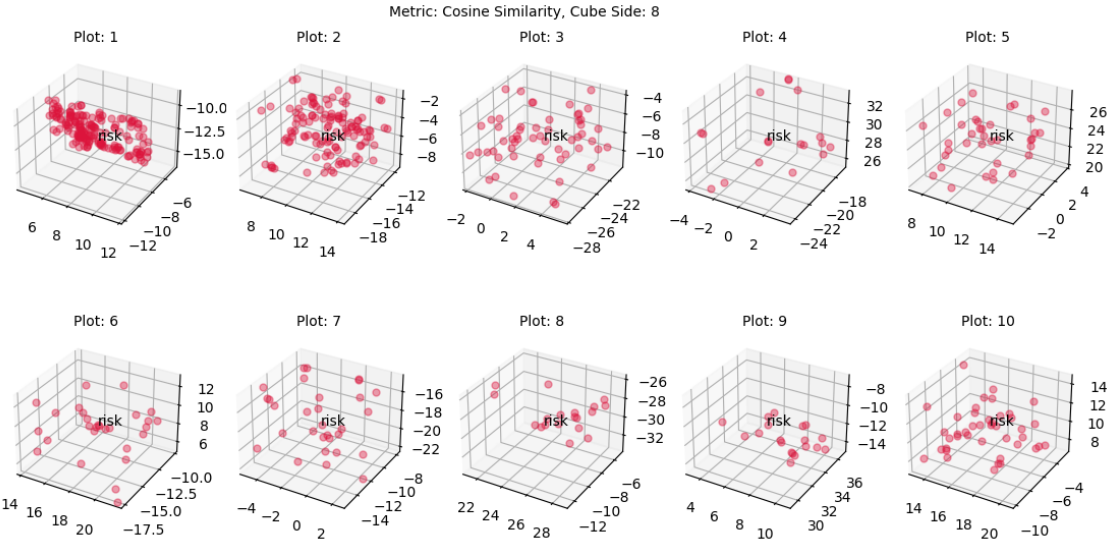


Figura B.1: Gráficos locais centrados em *risk* (*Corpus* 2; Divisão: 10; Volume: 512 u.v.; Redução: Similaridade de Cosseno).

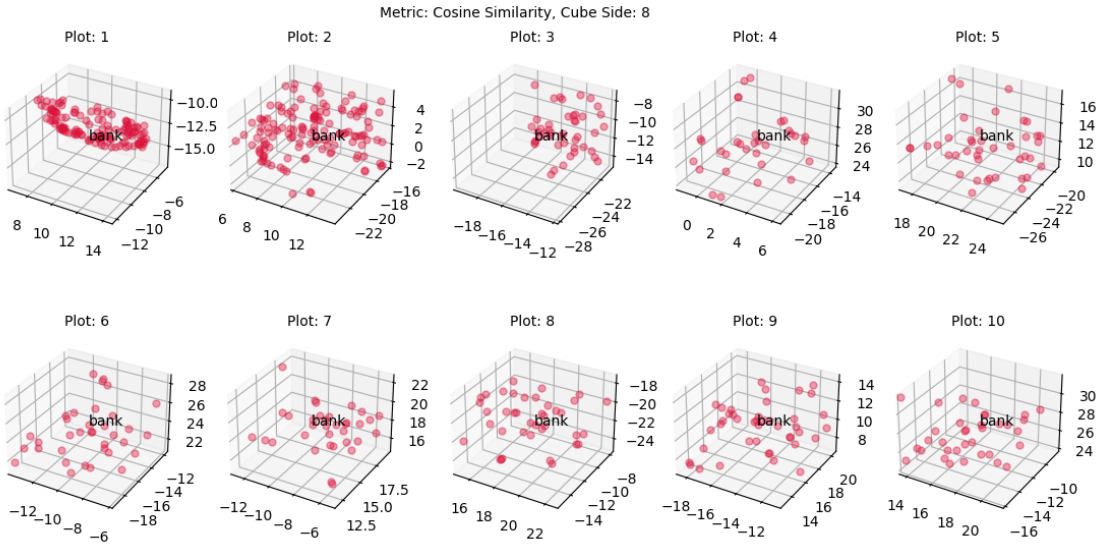


Figura B.2: Gráficos locais centrados em *bank* (*Corpus* 2; 10; 512 u.v.; Similaridade de Cosseno).

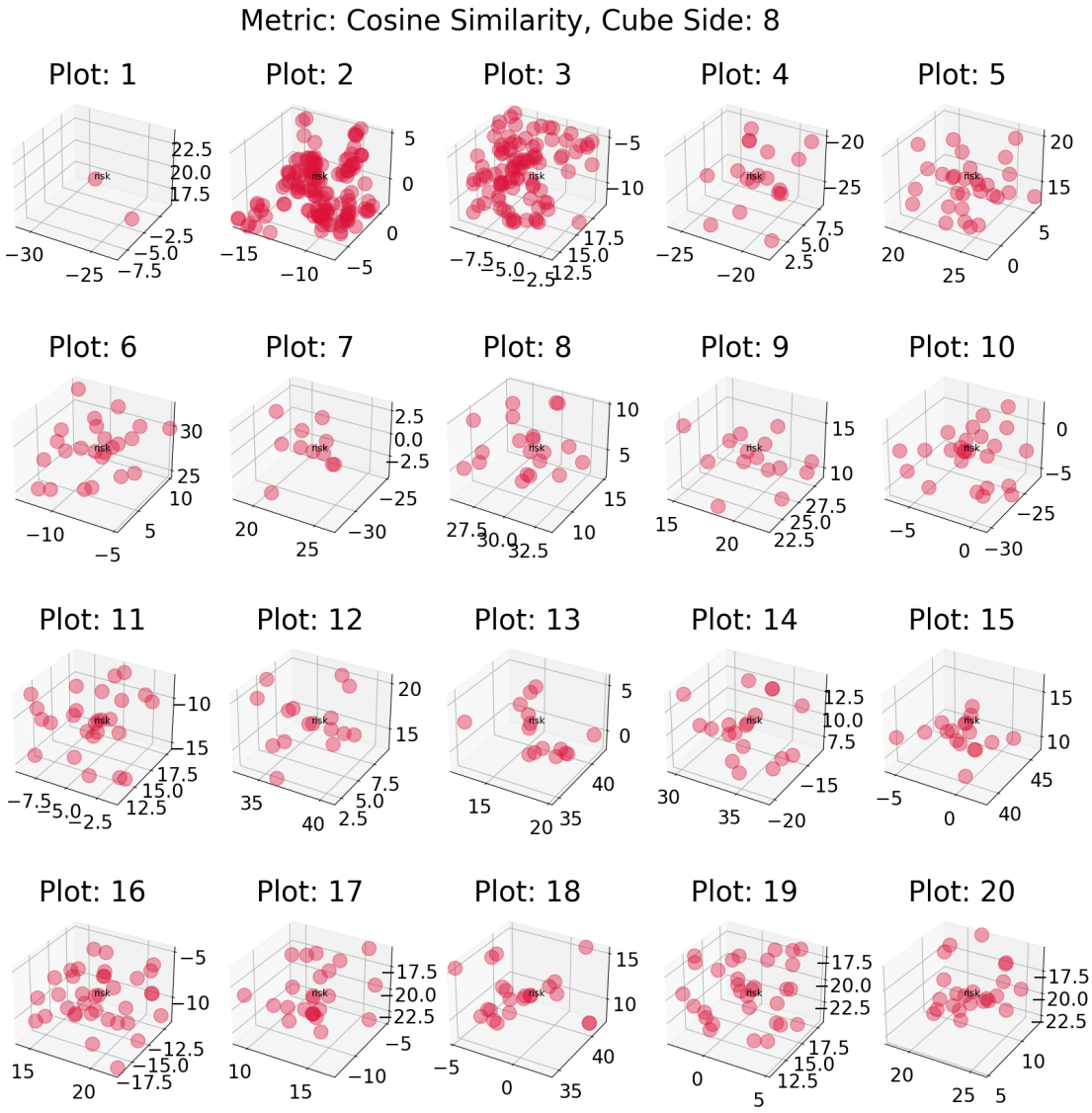


Figura B.3: Gráficos locais centrados em *risk* (*Corpus* 2; 20; 512 u.v.; Similaridade de Cosseno).

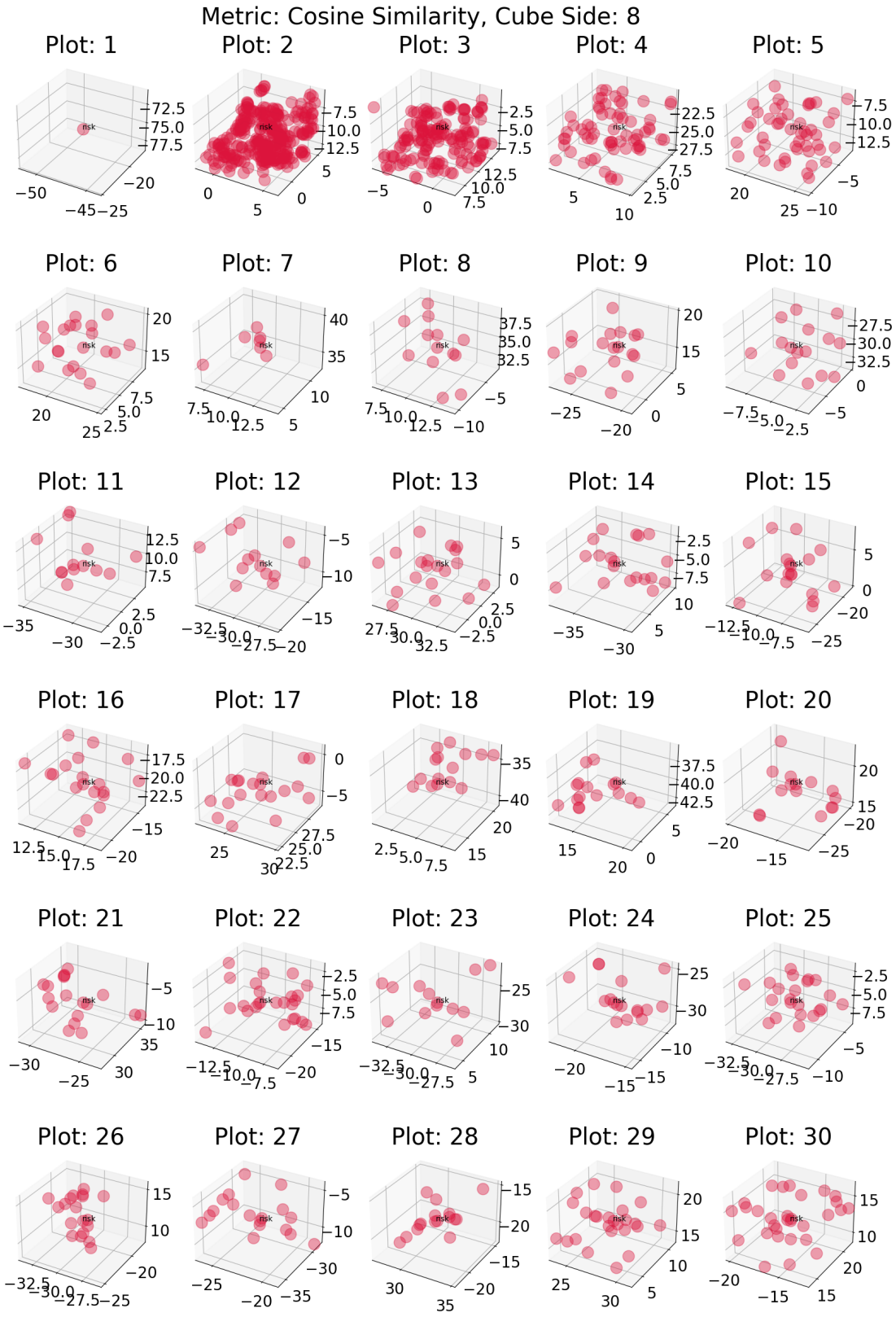


Figura B.4: Gráficos locais centrados em *risk* (*Corpus* 2; 30; 512 u.v.; Similaridade de Cosseno).

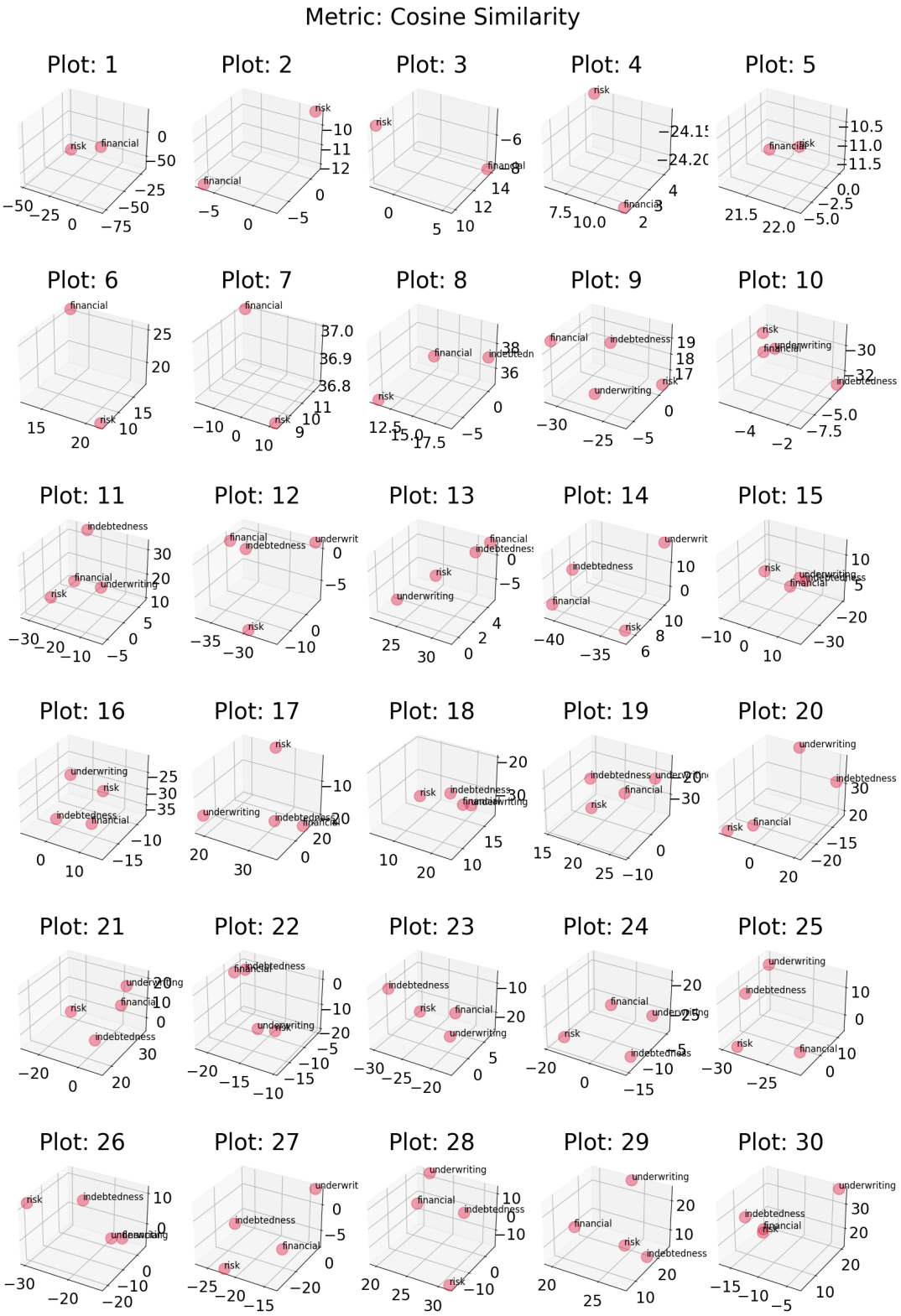


Figura B.5: Gráficos locais com as palavras cuja similaridade de cosseno com a palavra *risk* é maior que 0.45.