

Universidade Federal do Rio Grande do Sul
Instituto de Matemática e Estatística
Programa de Pós-Graduação em Estatística

Do embedding ao grafo: modelando partidas de futebol com BERT e NetLSD

Juliana Sena de Souza

Porto Alegre, Março de 2026.

CIP - Catalogação na Publicação

de Souza, Juliana Sena
Do embedding ao grafo: modelando partidas de futebol com BERT e NetLSD / Juliana Sena de Souza. -- 2026.
83 f.
Orientadora: Márcia Helena Barbian.

Dissertação (Mestrado) -- Universidade Federal do Rio Grande do Sul, Instituto de Matemática e Estatística, Programa de Pós-Graduação em Estatística, Porto Alegre, BR-RS, 2026.

1. Processamento de Linguagem Natural. 2. Embeddings. 3. GloVe. 4. BERT. 5. NetLSD. I. Barbian, Márcia Helena, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

Dissertação submetida por Juliana Sena de Souza como requisito parcial para a obtenção do título de Mestre em Estatística pelo Programa de Pós-Graduação em Estatística da Universidade Federal do Rio Grande do Sul.

Orientador(a):

Prof. Dra. Márcia Helena Barbian

Comissão Examinadora:

Prof. Dr. Danilo Marcondes Filho (PPGEst - UFRGS)

Prof. Dr. Marcos Oliveira Prates (PPGE - UFMG)

Prof. Dr. Rodrigo Citton Padilha dos Reis (PPGEpi - UFRGS)

Prof. Drª. Viviane Pereira Moreira (PPGC - UFRGS)

Data de Apresentação: 02 de Março de 2026

"Sometimes there's beauty in the tough words– it's all in how you read them."
- Jennifer Niven, All the Bright Places

Agradecimentos

Agradeço, antes de tudo, ao amor da minha vida: minha mãe. A mulher mais incrível que eu conheço, que me criou — a mim e aos meus irmãos — de forma tão excepcional que uma vida inteira ainda seria pouco para retribuir. Obrigada por ter sido base, força e cuidado, sempre.

Ao meu irmão gêmeo, que está comigo nessa vida desde os meus cinco minutos de vida e segue aqui, presente em tudo. Tu és sinônimo de pessoa e de amigo. Não sei o que eu seria sem ti — tu és metade de mim.

Aos meus sobrinhos, que eu amo profundamente e para quem eu desejo um futuro cheio de possibilidades. Que vocês tenham acesso a oportunidades tão incríveis quanto as que eu tive ao longo desses quase 11 anos na UFRGS, especialmente no caminho dos estudos.

À UFRGS, pela qualidade do ensino e pela formação que ultrapassa a sala de aula: pela oportunidade de conviver com pessoas brilhantes, de aprender com rigor, de desenvolver senso crítico e de me tornar profissional e pessoa nesse processo. Levo comigo não só conhecimento, mas também vivências que marcaram a minha trajetória.

Ao meu namorado, Leandro, pela vida que construímos juntos, por estar ao meu lado todos os dias, por ser parceria e por tornar a rotina leve com a tua presença — rindo comigo e sendo bobo junto, do nosso jeito. E aos nossos filhos-pet, que eu amo incondicionalmente e que fazem meus dias mais felizes com cada carinha fofa dormindo: Roma, Chico, Chef, Pandora, Ravena, Noel e a mais nova integrante da família, a Isa.

Aos meus amigos — os que a vida me deu ao longo do caminho, pessoalmente, pela internet, na faculdade e no trabalho. Eu sinceramente não sei como vocês me aguentam, mas ainda bem que aguentam. Obrigada por estarem por perto, por torcerem por mim e por serem parte de quem eu sou.

À minha orientadora, Márcia Barbian, por ter me guiado nesta etapa com generosidade e firmeza. Desde Inferência A seguimos juntas, e eu sou muito grata por essa parceria de anos, pela confiança e pelo apoio ao longo de todo o processo.

Aos membros da banca — Danilo, Marcos, Rodrigo e Viviane — obrigada por aceitarem fazer parte deste trabalho e por dedicarem tempo e atenção à leitura e às contribuições. Sou muito agradecida pela disponibilidade e pelo cuidado.

Resumo

A análise de partidas de futebol a partir de dados tabulares de eventos é amplamente utilizada, a partir desses dados é possível calcular número de passes certos, passes errados, quantidade de gols, quantidade de faltas, entre outros. Entretanto, tais métricas descritivas apresentam limitações quando o objetivo é comparar, de forma sistemática, padrões de jogo entre equipes, dado o caráter sequencial e contextual desses registros. Neste trabalho, propõe-se uma metodologia que integra Processamento de Linguagem Natural (NLP) e teoria dos grafos para transformar dados evento-a-evento em representações textuais e relacionais, permitindo extrair assinaturas do campeonato e quantificar similaridades entre os clubes. Inicialmente, os eventos são convertidos em um corpus textual por meio de templates (data-to-text generation), a partir do qual são obtidas representações vetoriais com modelos de linguagem, com ênfase em embeddings derivados do GloVe (Global Vectors for Word Representation) e BERT (Bidirectional Encoder Representations from Transformers). Medidas de proximidade baseadas em similaridade de cosseno são exploradas, com apoio de redução de dimensionalidade via UMAP (Uniform Manifold Approximation and Projection) e procedimentos de agrupamento via HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) para inspeção do espaço de representações. Em seguida, cada equipe é modelada como uma rede não direcionada e ponderada, em que nós representam jogadores e arestas codificam a similaridade entre vetores associados aos atletas, com filtragem por backbone para reduzir densidade e favorecer interpretação. A análise das redes inclui métricas de centralidade para identificar jogadores estruturalmente relevantes, e a comparação global entre equipes é realizada via descritor espectral NetLSD, seguido de distâncias entre assinaturas e agrupamento hierárquico. Os resultados indicam agrupamentos consistentes entre equipes e casos singulares, sugerindo que a abordagem captura regularidades compatíveis com diferentes formas de organização coletiva em campo.

Palavras-chave: Futebol; Processamento de Linguagem Natural; GloVe; BERT; *Embeddings*; Grafos; NetLSD.

Abstract

Soccer match analysis based on tabular event data is widely used, as this data enable the calculation of metrics such as successful passes, unsuccessful passes, number of goals, number of fouls, among others. However, such descriptive metrics present limitations when the objective is to systematically compare gameplay patterns across teams, given the sequential and contextual nature of these records. This study proposes a methodology that integrates Natural Language Processing (NLP) and graph theory to transform event-by-event data into textual and relational representations, enabling the extraction of league signatures and the quantification of similarities among clubs. Initially, events are converted into a textual corpus through templates (data-to-text generation), from which vector representations are obtained using language models, with emphasis on embeddings derived from GloVe (Global Vectors for Word Representation) and BERT (Bidirectional Encoder Representations from Transformers). Proximity measures based on cosine similarity are explored, supported by dimensionality reduction through UMAP (Uniform Manifold Approximation and Projection) and clustering procedures using HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) to inspect the representation space. Subsequently, each team is modeled as an undirected weighted network, in which nodes represent players and edges encode the similarity between vectors associated with athletes, with backbone filtering applied to reduce network density and improve interpretability. Network analysis includes centrality measures to identify structurally relevant players, while the global comparison between teams is performed using the NetLSD spectral descriptor, followed by distances between signatures and hierarchical clustering. The results indicate consistent grouping patterns among teams as well as singular cases, suggesting that the proposed approach captures regularities compatible with different forms of collective organization on the field.

Keywords: Football; Natural Language Processing; GloVe; BERT; *Embeddings*; Graphs; NetLSD.

Índice

1 Introdução **3**

2 Processamento de Linguagem Natural **8**

2.1 A história do Processamento de Linguagem Natural 8

2.2 Análise Textual 10

2.2.1 Representações distribuídas 11

2.2.1.1 Global Vectors for Word Representation (GloVe) 11

2.2.2 Representação de linguagem universal 13

2.2.2.1 *Transformers* 14

2.2.2.2 *Bidirectional Encoder Representations from Transformers* (BERT) . . 18

2.3 Redução de dimensionalidade 21

2.4 Similaridade entre palavras 22

2.5 Agrupamento de palavras 23

3 Grafos **25**

3.1 Definições básicas 25

3.2 Representações matriciais e Laplaciano do grafo 27

3.3 Comparação estrutural entre grafos 27

3.3.1 Descritor espectral NetLSD 28

4 Metodologia **29**

4.1 Base de dados 30
4.2 Preparação e pré-processamento dos dados 30
4.3 Visão geral da metodologia 33
4.4 Conversão de dados tabulares em texto 33
4.5 Extração e análise dos *embeddings* 34
4.5.1 Modelo GloVe 34
4.5.2 Modelo BERT 35
4.6 Análise das redes ponderadas 37
4.7 UMAP e HDBSCAN para inspeção dos embeddings contextualizados 39

5 Resultados **40**
5.1 Modelo GloVe 40
5.2 Modelo BERT 48

6 Conclusões e trabalhos futuros **60**

Anexos **69**

A Zonas da segmentação do campo **70**

B Distribuição das métricas de grau e centralidade, GloVe **71**

C Distribuição das métricas de grau e centralidade, BERT **73**

D Escolha de k para o agrupamento das equipes, GloVe **75**

E Escolha de k para o agrupamento das equipes, BERT **76**

Capítulo 1

Introdução

O futebol é um dos esportes mais populares do mundo e reúne uma ampla e dedicada base de torcedores, admiradores e simpatizantes em diferentes contextos. Sua relevância ultrapassa as quatro linhas do campo, manifestando-se também em dimensões econômicas e sociais. A magnitude dessa indústria pode ser observada na geração de receitas oriundas de múltiplas fontes. De acordo com a *Revisão Anual das Finanças do Futebol 2023* da Deloitte, a temporada de 2021/2022 das ligas europeias atingiu um recorde de receita de €29.5 bilhões. Entre as principais fontes de renda dos clubes, destacam-se os direitos de transmissão, a venda de ingressos e os patrocínios (Deloitte's Sports Business Group, 2023). Além do impacto direto no mercado esportivo, o futebol também se relaciona à mobilização de fluxos econômicos em regiões que sediam partidas e eventos de grande porte, incluindo efeitos associados ao turismo e ao consumo local (Tobar and Ramshaw, 2022; Oliveira et al., 2021).

Nesse contexto, a modelagem de dados de futebol é relevante por permitir uma compreensão mais profunda do jogo, apoiando análises que vão além de medidas tradicionais e observações estritamente descritivas. A disponibilidade de dados associados a partidas, equipes e jogadores possibilita a extração de evidências sobre desempenho coletivo e individual, padrões de jogo e estratégias adotadas ao longo do tempo, com aplicações em diferentes frentes da análise esportiva (Pantzalis and Tjortjis, 2020; Li et al., 2020; Yi et al., 2020, 2019; Teixeira et al., 2021).

Nos estágios iniciais, a análise de dados no futebol baseava-se majoritariamente no registro de eventos discretos ocorridos durante as partidas, como gols, chutes e passes, conhecidos como *event data*. A expansão dessa área foi fortemente influenciada por metodologias desenvolvidas em outros esportes, como beisebol, futebol americano e basquete (Fernández, 2021). Paralelamente, os avanços em capacidade computacional e armazenamento de dados viabilizaram o uso de modelos mais sofisticados, ampliando as possibilidades de avaliação técnica de jogadores, análise da dinâmica das partidas e previsão de resultados (Spearman, 2018). Ainda assim, abordagens tradicionais baseadas exclusivamente em estruturas tabulares e em agregações podem ter dificuldade em capturar relações complexas e contextuais inerentes ao jogo, especialmente quando o interesse está em padrões dependentes de sequência, coocorrência de ações e interação entre jogadores ao longo do tempo (Mendes-Neves et al., 2024b).

Nos últimos anos, o Processamento de Linguagem Natural (NLP) consolidou-se como um campo central na análise de dados textuais e sequenciais, com avanços impulsionados por novas formas

de representação de linguagem. Um marco relevante nesse processo foi a adoção de representações distribuídas (Bengio et al., 2003), nas quais palavras e unidades linguísticas são mapeadas para vetores em espaços de alta dimensão, permitindo que similaridades geométricas reflitam padrões de uso e associação. Entre essas abordagens, métodos como o GloVe *(Global Vectors for Word Representation)* estimam vetores a partir de estatísticas globais de coocorrência, resultando em *embeddings* estáticos que sintetizam regularidades do *corpus* (Pennington et al., 2014). Apesar de úteis como ponto de partida, *embeddings* estáticos atribuem uma única representação para cada *token*, o que limita a captura de polissemia e de variações de sentido condicionadas ao contexto.

A partir dessa limitação, abordagens contextualizadas passaram a ocupar papel central no estado da arte. Modelos baseados em arquiteturas *Transformer* introduziram mecanismos de atenção capazes de incorporar dependências de longo alcance e representar *tokens* de acordo com o contexto em que aparecem (Vaswani et al., 2017). Nesse cenário, o BERT *(Bidirectional Encoder Representations from Transformers)* tornou-se uma referência ao propor um pré-treinamento bidirecional baseado em modelagem mascarada, permitindo obter *embeddings* contextualizados que se adaptam ao uso local da linguagem e podem ser explorados em diferentes tarefas (Devlin et al., 2019). De forma geral, tais modelos ampliaram a capacidade de modelar relações semânticas e estruturais em sequências, mas também introduziram desafios metodológicos relevantes, como o limite de comprimento de entrada e a necessidade de decisões sobre segmentação e interpretação de representações em diferentes níveis.

No contexto do futebol, técnicas de Processamento de Linguagem Natural têm sido exploradas em diferentes frentes. Uma primeira linha de pesquisa concentra-se na geração automática de textos a partir de dados estruturados das partidas, como a produção de comentários e descrições de jogos com base em sequências de eventos (Taniguchi et al., 2019), bem como no desenvolvimento de plataformas voltadas à geração de resumos textuais a partir de estatísticas de jogo (Fernandes et al., 2024).

Em paralelo, observa-se um movimento mais recente que busca aproximar a modelagem de partidas da lógica de "linguagem", tratando eventos esportivos como unidades análogas a *tokens* e adotando objetivos inspirados em modelos de linguagem para capturar dependências temporais e contextuais (Mendes-Neves et al., 2024b,a; Romero et al., 2026). Nesse sentido, o modelo RisingBALLER (Adjileye, 2024) propõe um arcabouço baseado em *transformers* treinado sobre dados de eventos da StatsBomb, no qual as representações dos jogadores são aprendidas de forma contextualizada, levando em conta simultaneamente os elencos em campo, as estatísticas da partida, as posições e as equipes. O modelo é pré-treinado por meio de uma tarefa auto-supervisionada de predição de jogadores e avaliado em tarefas *downstream*, como a previsão de estatísticas da próxima partida.

Outra linha relacionada, embora conceitualmente distinta, é explorada por Horton and Lucey (2025), que propõem um modelo baseado em *transformers* axiais para realizar previsões em tempo real dos totais de múltiplas ações em partidas de futebol, em níveis de jogador, equipe e jogo. O método utiliza um conjunto de 12 estatísticas de ações da partida: gols, assistências, finalizações, passes certos e errados, escanteios, desarmes, faltas, cartões e gols sofridos; combinadas com contagens acumuladas do estado do jogo (tempo, placar e tipo de evento) e estatísticas históricas pré-jogo que representam a força de jogadores e equipes.

Em um domínio distinto, Zhang and Yang (2025) introduzem o *Football Transformer* (FoT), um modelo baseado em *transformers* no contexto de visão computacional. Nesse caso, o método opera sobre mapas de características espaciais extraídos de imagens RGB por meio de *Vision Transformers*,

com o objetivo de realizar detecção de objetos em tempo real. Diferentemente de abordagens inspiradas em NLP, o FoT não modela eventos esportivos como sequências simbólicas nem utiliza *tokens* ou dependências semânticas linguísticas, mas sim relações espaciais e visuais contínuas entre pixels e regiões da imagem.

De forma semelhante, o trabalho de Capellera et al. (2024) propõe o FootBots, um modelo baseado em *Transformers* para a predição de trajetórias de jogadores e da bola, que atua diretamente sobre trajetórias espaço-temporais contínuas e interações entre agentes. Apesar de empregar a arquitetura *Transformer*, o FootBots também não trata os dados como texto, operando exclusivamente sobre coordenadas espaciais e rótulos de agentes, sem qualquer noção de semântica linguística.

Em conjunto, esses trabalhos sugerem que modelagens sequenciais e contextuais, baseadas em mecanismos de atenção, oferecem uma alternativa promissora e escalável às representações manuais e às estatísticas agregadas fixas tradicionalmente utilizadas na análise de futebol. Essas abordagens reforçam a ideia de que representações sequenciais podem preservar informações relevantes que tendem a se perder quando os dados são analisados exclusivamente por meio de agregações tabulares.

Apesar do avanço nessas técnicas e do crescente interesse por abordagens orientadas a dados no futebol, bases detalhadas de eventos ainda são, em grande medida, produzidas e comercializadas por fornecedores privados (*e.g.*, Opta, Wyscout, StatsBomb), o que restringe o acesso amplo a informações essenciais para análises independentes e reprodutíveis (Pappalardo et al., 2019). Mesmo quando versões abertas são disponibilizadas, elas tendem a representar exceções em relação à base de dados mais completa utilizada na prática profissional (Pappalardo et al., 2019).

Por fim, além das representações estritamente textuais, o futebol também pode ser descrito como um sistema relacional, no qual interações entre jogadores e eventos induzem estruturas em rede. Assim, abordagens baseadas em grafos têm sido utilizadas para complementar análises centradas em sequência, permitindo estudar propriedades estruturais de equipes e partidas, além de capturar padrões não observáveis por métricas individuais (Buldú et al., 2018). Nesse contexto, descritores espectrais fornecem uma forma de sumarizar propriedades globais do grafo em diferentes escalas. O NetLSD *(Network Laplacian Spectral Descriptor)*, por exemplo, propõe uma assinatura multiescala baseada no espectro do grafo, adequada para tarefas de comparação e similaridade estrutural (Tsitsulin et al., 2018).

Dessa forma, a combinação de *embeddings* estáticos, que capturam regularidades globais de co-ocorrência (*e.g.*, GloVe), com *embeddings* contextualizados, capazes de incorporar dependências de contexto ao longo de sequências (*e.g.*, BERT), e com descritores estruturais de grafos, que sumarizam propriedades globais em múltiplas escalas (*e.g.*, NetLSD), oferece um arcabouço consistente para explorar o futebol como um sistema complexo, no qual texto, sequência e relações estruturais podem ser analisados de maneira integrada (Pennington et al., 2014; Devlin et al., 2019; Tsitsulin et al., 2018).

Objetivo

Objetivo Geral

Desenvolver e aplicar uma metodologia estatística que transforme dados tabulares de eventos de

futebol em representações textuais e vetoriais, combinando técnicas de NLP e análise de grafos para caracterizar padrões de jogo comparáveis entre equipes.

Objetivos Específicos

- **Converter os dados tabulares de partidas em texto** (*corpus*) por meio de *templates*, permitindo o uso de modelos de linguagem na análise de sequências de eventos.
- **Gerar representações vetoriais** para o *corpus* por meio de *embeddings* e, quando necessário, agregar representações para viabilizar comparações.
- **Explorar a organização das representações** em baixa dimensão utilizando UMAP e destacar estruturas locais por meio de agrupamento via HDBSCAN, com foco na análise de partidas.
- **Construir, para cada equipe, uma rede não direcionada e ponderada** em que os nós representam jogadores e os pesos das arestas são definidos pela similaridade de cosseno entre os vetores associados aos atletas.
- **Identificar jogadores-chave em cada equipe** usando métricas de centralidade na rede de similaridade, como o grau ponderado (*strength*) e a centralidade de intermediação (*betweenness*), interpretando-as como papéis complementares.
- **Comparar estruturalmente as equipes** por meio de distâncias baseadas em NetLSD, construindo uma matriz de similaridades e aplicando agrupamento hierárquico para obter grupos de clubes com padrões estruturais semelhantes.

Suporte computacional

As análises foram realizadas no `Python` versão 3.11.5 e no `R` versão 4.2.2.

Organização do trabalho

Esta dissertação está organizada em seis capítulos. No Capítulo 2 é apresentada a fundamentação teórica relacionada ao **Processamento de Linguagem Natural**, contemplando sua evolução histórica, as principais abordagens de análise textual e técnicas associadas, bem como a descrição dos modelos **GloVe** e **BERT**. Ainda nesse capítulo, são discutidas estratégias para representação e visualização de *embeddings*, com destaque para o **UMAP**, a **similaridade de cosseno** como medida de proximidade e o **HDBSCAN** como alternativa de agrupamento. No Capítulo 3, são apresentados os fundamentos da **Teoria dos Grafos** e da análise espectral de redes, incluindo o Laplaciano do grafo e métodos de comparação estrutural, com ênfase no descritor espectral **NetLSD**, adotado para comparar redes, potencialmente de tamanhos distintos, no contexto desta dissertação. No Capítulo 4 é detalhada a metodologia proposta, abrangendo a preparação dos dados, a construção do *corpus* a partir de eventos de jogo, a extração e análise de *embeddings* e a construção das redes de jogadores baseadas em similaridade, bem como as estratégias de comparação e agrupamento entre equipes. O Capítulo 5

apresenta os resultados obtidos, juntamente com sua análise e discussão, incluindo visualizações, medidas de similaridade e interpretações dos padrões encontrados. Por fim, o Capítulo 6 reúne as conclusões do trabalho, além de limitações, contribuições e perspectivas para trabalhos futuros.

Convenções de notação

Esta dissertação utiliza notação definida localmente por capítulo e/ou subseção. Em particular, alguns símbolos podem aparecer em mais de um contexto com significados distintos (por exemplo, w em modelos de *embeddings* como GloVe e em modelos do tipo *transformer*; ou $d(\cdot, \cdot)$ em procedimentos de redução de dimensionalidade, medidas de distância e construção/análise de grafos). Essa escolha foi adotada por alinhar a escrita às convenções mais recorrentes em cada área e para favorecer a interpretação. Em todos os casos, as variáveis e funções são explicitamente definidas no ponto em que são utilizadas.

Capítulo 2

Processamento de Linguagem Natural

Neste capítulo, é apresentada inicialmente uma revisão sobre **Processamento de Linguagem Natural** (NLP), contemplando sua história e evolução desde os primeiros sistemas baseados em regras até as abordagens modernas. Em seguida, é discutido o conceito de análise textual, com detalhamento das diferentes abordagens existentes, suas quatro categorias e as principais técnicas associadas a cada uma delas. Nesse contexto, são apresentados os modelos de linguagem **GloVe** (Vetores Globais para Representações de Palavras, do inglês *Global Vectors for Word Representation*) e **BERT** (Representações de Codificador Bidirecional de Transformadores, do inglês *Bidirectional Encoder Representations from Transformers*), baseado na arquitetura **Transformers**, ambos adotados como referência metodológica nesta dissertação. Além disso, serão abordadas seções relativas à visualização de *embeddings*, com destaque para a técnica de redução de dimensionalidade **UMAP** (*Uniform Manifold Approximation and Projection*) e para a noção de **similaridade de cosseno** como medida de proximidade entre representações vetoriais. Por fim, será apresentado o **HDBSCAN** (*Hierarchical Density-Based Spatial Clustering of Applications with Noise*) como alternativa para o agrupamento de palavras.

2.1 A história do Processamento de Linguagem Natural

O Processamento de Linguagem Natural (NLP) é um campo da ciência da computação e linguística, comumente associado a uma subárea da inteligência artificial (IA) cuja ênfase está nas interações entre computadores e a linguagem humana. Em NLP, as técnicas são desenvolvidas com o objetivo de que o computador entenda os comandos em linguagem natural e atue de acordo com essas informações (Kumar, 2013).

O NLP surgiu no cenário das primeiras investigações em IA, cujo propósito era habilitar as máquinas a entender e produzir a linguagem humana de forma relevante. Frequentemente, o início do NLP está ligado ao artigo seminal de Alan Turing, "*Computing Machinery and Intelligence*" (Turing, 1950). Neste estudo, Turing sugeriu o que hoje é conhecido como o **Teste de Turing**, que avaliava a habilidade de uma máquina de demonstrar comportamento que se assemelha ao de um humano durante uma conversa natural.

Na década de 1950 e início da década de 1960, os primeiros esforços no campo estavam mais

focados na **tradução automática**, quando havia grande interesse em traduzir textos técnicos entre idiomas, especialmente entre o inglês e o russo. Os primeiros experimentos marcaram o início da aplicação de computadores às ciências da linguagem e à tradução automática, embora uma das tarefas mais desafiadoras no NLP fosse a primeira aplicação não digital dos computadores, concebida como uma tecnologia de guerra para fornecer traduções em massa para os propósitos estratégicos da Guerra Fria (Léon, 2014).

O experimento de *Georgetown* em 1954 foi capaz de realizar a tradução automática de mais de 60 frases do russo para o inglês. Apesar do entusiasmo inicial, com muitas previsões de progressos iminentes e sistemas operando totalmente automatizados em alguns anos, a complexidade inerente aos desafios linguísticos começou a se tornar mais evidente. Esses sistemas iniciais de tradução eram bastante limitados, operando com um vocabulário restrito e incapazes de lidar com a ambiguidade da linguagem natural, por exemplo. Essas limitações resultaram no famoso relatório ALPAC (do inglês *Automatic Language Processing Advisory Committee*), que concluiu que a tradução automática era mais lenta, menos precisa e mais cara do que a tradução humana, o que levou ao desaceleramento dos investimentos em tradução automática nos Estados Unidos (Hutchins, 1995).

Durante as décadas de 1960 e 1970, diversos sistemas de NLP foram desenvolvidos, como por exemplo o SHRDLU (Winograd, 1972), um sistema que compreendia comandos simples, como responder perguntas e executar ordens, com um vocabulário restrito conhecido como "mundos de blocos" (do inglês *block worlds*), e o ELIZA (Weizenbaum, 1966), o primeiro programa de NLP desenvolvido, um *chatbot* que simulava um psicoterapeuta. Nos anos 1970, foi introduzido o conceito de ontologias conceituais, que organizavam o conhecimento em redes de conceitos e relações, permitindo que os computadores processassem e entendessem informações de forma mais próxima do raciocínio humano. Sistemas como o MARGIE (Schank, 1975) e o SAM (Cullingford, 1977) são exemplos.

Até os anos 1980, os sistemas de processamento de linguagem natural dependiam de regras manuais, mas, a partir do final da década, os algoritmos de aprendizado de máquina começaram a transformar essa área. Os anos de 1990 trouxeram uma transição significativa para **métodos estatísticos** no processamento de linguagem natural, impulsionada pela crescente disponibilidade de grandes *corpora* de dados textuais digitais e pelo avanço nas técnicas computacionais. Conforme discutido por Church and Mercer (1993), essa transformação foi possível graças à implementação de modelos probabilísticos, fundamentados na teoria da informação, que permitiram ao NLP lidar com a variabilidade e a incerteza inerentes à linguagem humana.

Estes novos métodos representaram uma ruptura com as metodologias anteriores fundamentadas em regras, que eram mais rígidas e menos eficazes para lidar com a variabilidade da fala, à medida que os algoritmos estatísticos começaram a dominar tarefas como o reconhecimento de fala e a tradução instantânea (Jelinek, 1990; Brown et al., 1993). Com a introdução de métodos como Máquinas de Vetores de Suporte (do inglês *Support Vector Machine*, SVM) (Cortes and Vapnik, 1995) e outros classificadores, o campo de NLP passou a realizar análises mais escaláveis e robustas, especialmente em tarefas como a classificação de texto. Ademais, o surgimento do *WordNet*, uma base de dados lexical criada por Miller (1995), desempenhou um papel crucial como recurso de conhecimento linguístico estruturado, auxiliando no progresso da compreensão semântica em sistemas de Processamento de Linguagem Natural.

A década de 2010 marcou o início da era do *deep learning* no NLP, em que as redes neurais

foram sendo aplicadas de maneira mais abrangente. Um marco significativo foi a criação do *word2vec* (Mikolov et al., 2013), que permitiu a geração de representações distribuídas de palavras por meio de vetores densos, capturando a semântica de maneira mais eficaz do que as técnicas anteriores. Em seguida, as Redes Neurais Recorrentes (RNNs) e os modelos *seq2seq* (Sutskever et al., 2014), combinados com mecanismos de atenção (Bahdanau et al., 2014), trouxeram avanços substanciais para tarefas como tradução automática e síntese de texto. No entanto, a verdadeira revolução ocorreu com a introdução da arquitetura Transformer (Vaswani et al., 2017), que se estabeleceu como a base para modelos pré-treinados mais poderosos.

Por fim, os anos 2020 marcaram a era dos modelos de grande escala (conhecidos como LLMs, ou *Large Language Models*, em inglês), com o lançamento de modelos como BERT (*Bidirectional Encoder Representations from Transformers*) (Devlin et al., 2019), T5 (*Text-To-Text Transfer Transformer*) (Raffel et al., 2019), GPT-3 (*Generative Pre-trained Transformer 3*) (Brown et al., 2020) e PaLM (*Pathways Language Model*) (Chowdhery et al., 2022). Esses modelos trouxeram grandes melhorias em múltiplas tarefas de NLP, incluindo tradução automática, resposta a perguntas e geração de texto. Esses avanços estabeleceram um novo padrão de qualidade, com GPT-3 e PaLM destacando-se pela capacidade de lidar com tarefas complexas de linguagem natural, enquanto o BERT e T5 abriram novos caminhos para o pré-treinamento e ajuste fino em tarefas específicas de compreensão e tradução.

2.2 Análise Textual

A representação textual é a transformação dos dados de entrada, em um formato numérico, seja vetor ou matriz. A representação textual como vetores numéricos é conhecida como modelo de espaço vetorial (*Vector Space Model*, VSM), um modelo algébrico utilizado para representar qualquer conjunto de texto (Vajjala et al., 2020).

A transformação dos dados brutos em um formato que pode ser utilizado efetivamente por algoritmos de aprendizado de máquina é um passo necessário em qualquer conjunto de dados, seja este imagens, textos ou fala. Em processamento de imagens, a transformação da imagem em um conjunto numérico que servirá como *input* no modelo é definido por uma matriz de *pixels*, cuja representação é dada pela intensidade do *pixel* correspondente da imagem. Similarmente, o processamento de vídeos considera que cada *frame* do vídeo é uma imagem, portanto pode ser considerado como uma sequência de matrizes, cada uma representando um *frame* do vídeo. Entretanto, a representação textual não possui uma representação matemática tão direta (Vajjala et al., 2020).

Ainda, de acordo com Vajjala et al. (2020), as abordagens de representação textual podem ser organizadas em quatro categorias: abordagens básicas de vetorização, **representações distribuídas**, **modelos de linguagem universal** e características feitas manualmente.

Apesar de simples e rápidas, as técnicas básicas de vetorização (por exemplo, representações baseadas em contagens) produzem vetores esparsos e de alta dimensionalidade, cuja dimensão cresce com o tamanho do vocabulário, podendo ser computacionalmente ineficientes. Além disso, por tratarem palavras, ou n-gramas como unidades discretas, essas técnicas têm capacidade limitada de capturar relações semânticas e de lidar com termos fora do vocabulário (*out-of-vocabulary*). Por sua vez, abordagens baseadas em características feitas manualmente dependem de escolhas específicas do domínio e

de *feature engineering*, o que reduz a portabilidade entre tarefas (Vajjala et al., 2020). Como esta dissertação se concentra em modelos distribuídos e modelos de linguagem pré-treinados, não entraremos em detalhes nas abordagens básicas de vetorização e nas características feitas manualmente.

2.2.1 Representações distribuídas

Como forma de lidar com as principais desvantagens das técnicas citadas anteriormente, métodos de representações com baixa dimensionalidade foram desenvolvidos. Esses métodos utilizam arquiteturas baseadas em redes neurais para criar representações de palavras e textos mais densas (poucos zeros) e compactas (baixa dimensão) (Vajjala et al., 2020).

Neste contexto, Bengio et al. (2003) propuseram uma metodologia de representações distribuídas capaz de utilizar vários algoritmos diferentes na "aprendizagem" para a criação desses vetores numéricos. Essa metodologia é conhecida como **Word Embeddings** (ou incorporação de palavras). A ideia principal dos *word embeddings* é que palavras similares ocorrem em contextos semelhantes. Essas representações textuais capturam semelhanças distribucionais, ou seja, palavras com significados semelhantes tendem a aparecer em contextos semelhantes dentro de um texto.

Diversos modelos de *word embeddings* foram desenvolvidos ao longo dos anos, cada um com suas abordagens e vantagens específicas. Um dos primeiros e mais influentes modelos é o **Word2Vec**, introduzido por (Mikolov et al., 2013). O *Word2Vec* utiliza dois algoritmos principais: o *Continuous Bag of Words* (CBOW), que visa prever uma palavra com base em seu contexto; e o *Skip-gram*, que faz a previsão do contexto a partir de uma palavra.

O **FastText** (Bojanowski et al., 2016) é uma extensão do *Word2Vec* que introduz a modelagem de subpalavras. Enquanto o *Word2Vec* utiliza palavras inteiras para gerar vetores, o *FastText* considera $n-$gramas de caracteres em sua representação. Essa abordagem é baseada no *Skip-gram*, no qual cada palavra é representada por uma combinação dos vetores dos $n-$gramas que a compõem. Ao incorporar informações de subpalavras, o *FastText* melhora significativamente a representação de palavras raras, línguas morfologicamente ricas e conjuntos de dados de treinamento pequenos (Bojanowski et al., 2016) .

Outro modelo de representações de palavras é o **GloVe** (*Global Vectors for Word Representation*), proposto por Pennington et al. (2014). O GloVe é utilizado para capturar relações semânticas e sintáticas entre palavras de maneira eficaz através de matrizes de coocorrência global das palavras para aprender as representações vetoriais.

2.2.1.1 Global Vectors for Word Representation (GloVe)

O modelo GloVe (**Global Vectors**) é um modelo de incorporação de palavras introduzido em Pennington et al. (2014) utilizado para capturar relações semânticas e sintáticas entre palavras de maneira eficaz.

Considere **X** como sendo a contagem da matriz de coocorrência de palavras, em que X_{ij} representa

o número de vezes que a palavra j ocorre no contexto da palavra i e $X_i = \sum_k X_{ik}$ o número de vezes que qualquer palavra aparece no contexto da palavra i. A probabilidade da palavra j aparecer no contexto da palavra i é denotada por

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}.$$

O ponto de partida para definir o modelo considerou que taxas de probabilidades de coocorrência são melhores para distinguir tanto palavras relevantes das irrelevantes, quanto discriminar entre duas palavras relevantes, em comparação com as probabilidades brutas. Com isto, o modelo geral inicial foi definido como

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}, \tag{2.1}$$

que depende das palavras i, j e k, onde $w \in \mathbb{R}^d$ são vetores de palavras e $\tilde{w} \in \mathbb{R}^d$ são vetores de palavras de contexto, sendo d a dimensão do espaço de *embeddings*, isto é, o número de componentes de cada vetor. A função F na Equação 2.1 foi definida como uma função que depende apenas da diferença entre as duas palavras alvo w_i e w_j, ou seja

$$F(w_i{-}w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}. \tag{2.2}$$

Para evitar que F na Equação 2.2 seja uma função complicada que ofusque a estrutura linear, os autores incorporaram o produto escalar na parte esquerda da equação. Além disso, como a distinção entre a palavra e a palavra de contexto é arbitrária na matriz de coocorrência de palavras, elas podem ser trocadas entre si, ou seja $w \leftrightarrow \tilde{w}$ e $X \leftrightarrow X^T$. Basicamente, estamos alterando os *inputs* para F no lado esquerdo e desejamos ter um impacto semelhante no lado direito da Equação. Como a Equação 2.2 não é invariante a essa reformulação, a simetria foi reformulada em dois passos:

1. É necessário manter a similaridade estrutural entre os dois grupos (função homomórfica), então a subtração dos dois vetores no lado esquerdo da equação deve refletir a divisão desses vetores no lado direito. Isso sugere a concepção de que a representação vetorial das palavras deve manter as relações semânticas durante uma operação de subtração, da mesma forma que as relações de coocorrência são preservadas em uma operação de divisão. A equação resultante é definida por

   $$F((w_i{-}w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}, \tag{2.3}$$

   e que F é definida com a função exponencial, $F(x) = \exp(x)$, e como $F(w_i^T \tilde{w}_k) = P_{ik}$ então tomando o $\log$ em ambos os lados

   $$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i). \tag{2.4}$$

2. A Equação 2.4 exibe simetria em termos de palavra e contexto, exceto por $\log(X_i)$. Como esse termo é independente de w_k, pode-se considerar $\log(X_i)$ como um vício b_i para w_i e $\tilde{b}_k$ um vício para $\tilde{w}_k$, ou seja

   $$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}).$$

Para resolver os problemas remanescentes do modelo, como a má definição no logaritmo e o fato de o modelo definir igualmente os pesos de todas as coocorrências, foi proposto um modelo de regressão de mínimos quadrados ponderados:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2, \tag{2.5}$$

em que J é uma função de custo a ser minimizada sobre a matriz de coocorrência com tamanho de vocabulário V, w_i é o vetor de palavra, $\tilde{w}_k$ o vetor de contexto, X_{ij} é o termo da matriz de coocorrência e $f(X_{ij})$ é uma função ponderada que deve obedecer às seguintes propriedades:

1. $f(0) = 0$, então se f é vista como uma função contínua, $x \to 0$ rápido o suficiente para que $lim_{x\to 0} f(x) log^2 x$ seja finito;
2. $f(x)$ deve ser não-decrescente para que coocorrências raras não tenham pesos excessivos e
3. $f(x)$ deve ser relativamente pequena para valores altos de x, para que coocorrências frequentes não tenham pesos excessivos.

Por fim, os autores especificaram como função para $f(X_{ij})$ a seguinte parametrização:

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{se } x < x_{\max} \\ 1 & \text{caso contrário .} \end{cases}$$

Em comparação com o `word2vec` (Mikolov et al., 2013), considerando o mesmo *corpus*, vocabulário, tamanho de janela e tempo de treinamento, o GloVe consistentemente o supera. Além de alcançar os melhores resultados, estes são alcançados mais rapidamente (Pennington et al., 2014).

2.2.2 Representação de linguagem universal

Tanto as abordagens básicas de vetorização quanto as representações distribuídas atribuem uma representação fixa a cada palavra. No entanto, as palavras podem ter significados diferentes dependendo do contexto. Por exemplo, nas frases "eu fui ao banco para sacar dinheiro" e "eu sentei no banco da praça", a palavra "banco" aparece em ambos os casos, mas com sentidos distintos. Nessas técnicas, a palavra teria apenas uma única representação vetorial, sem uma maneira direta de capturar essas variações de significado causadas pelos diferentes contextos (Vajjala et al., 2020).

Para solucionar esse problema, as representações contextuais de palavras foram desenvolvidas. Essas técnicas utilizam modelagem de linguagem, que é a tarefa de predizer a próxima palavra provável em uma frase. Uma das primeiras e mais influentes abordagens foi o ELMo (*Embeddings from Language Models*), desenvolvido pela equipe da *Allen Institute for AI* (Peters et al., 2018). O ELMo utiliza redes neurais recorrentes (RNNs) com uma arquitetura de LSTM (Memória de Longo e Curto Prazo, do inglês *Long Short-Term Memory*) bidirecional para gerar as representações contextuais de palavras.

Outra técnica sensível ao contexto, ou seja, que produz diferentes representações para palavras com a mesma escrita, é o BERT (*Bidirectional Encoder Representations from Transformers*), um modelo de linguagem desenvolvido por pesquisadores do *Google AI* (Devlin et al., 2019). O BERT é baseado na arquitetura **Transformer**, o que permite processar todas as palavras de uma frase paralelamente.

2.2.2.1 Transformers

A estrutura do *Transformer* é baseada na arquitetura codificador e decodificador (*encoder-decoder*). Considere a seguinte notação, o codificador mapeia uma sequência de entrada $\mathbf{x} = (x_1, \ldots, x_n)$ de representações de símbolos para uma sequência $\mathbf{z} = (z_1, \ldots, z_n)$ de representações contínuas. Após, dado $\mathbf{z}$, o decodificador gera uma sequência de saída de símbolos $\mathbf{y} = (y_1, \ldots, y_n)$, um elemento por vez. Os símbolos gerados $\mathbf{y}$ são utilizados como entrada adicional do modelo ao gerar a próxima sequência de saída, pois em cada passo o modelo é autorregressivo. A Figura 2.1 apresenta a arquitetura do modelo *Transformer* (Vaswani et al., 2023). Abaixo é apresentado alguns conceitos sobre cada parte do modelo.

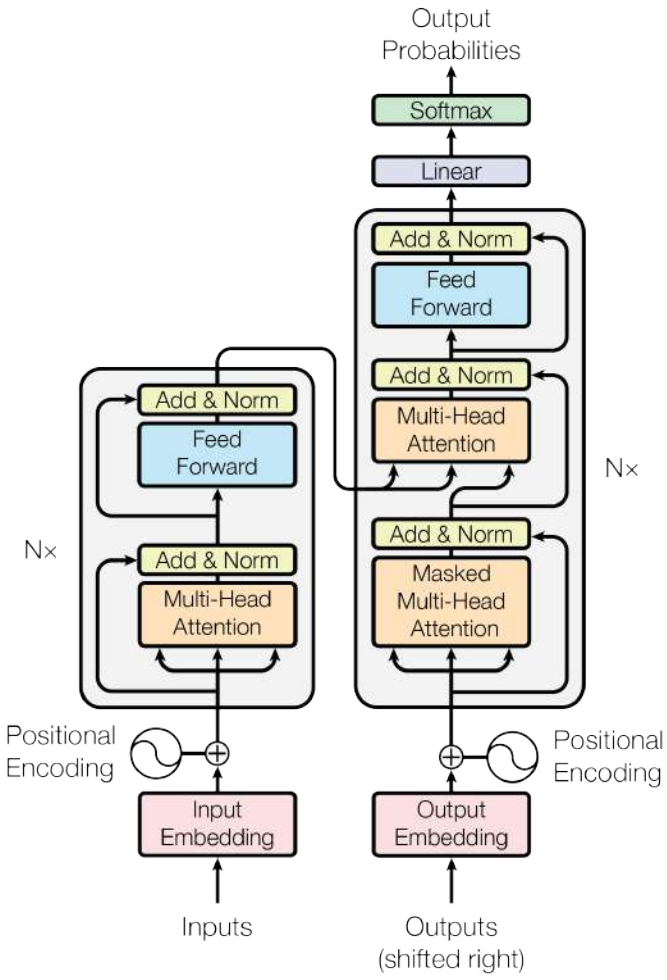


Figura 2.1: Arquitetura de um *Transformer*
Fonte: Vaswani et al. (2023)

Codificador e Decodificador

Codificador (Encoder):

No artigo original (Vaswani et al., 2023), o codificador é composto por um conjunto de codificadores de 6 camadas que possuem a mesma estrutura, sendo que cada camada possui duas subcamadas: o mecanismo de autoatenção com vários cabeçotes (*Multi-Head Attention*); e uma rede neural direta simples, posicional e totalmente conectada (*Feed Forward*). Ao redor de cada subcamada é empregada uma conexão residual, seguido por uma camada de normalização. Todas as subcamadas no modelo, assim como as camadas de *embeddings*, produzem saídas de dimensão $d_{model} = 512$ para facilitar as conexões residuais. A saída de cada subcamada é dada por

$$\text{LayerNorm}(x + \text{Sublayer}(x)),$$

em que x denota a representação de entrada da subcamada, isto é, os vetores ocultos associados aos tokens, em $\mathbb{R}^{d_{\text{model}}}$ e $\text{Sublayer}(\cdot)$ representa a transformação implementada em cada subcamada.

Decodificador (Decoder):

O decodificador, similar ao codificador, também foi originalmente proposto como uma composição de um conjunto de 6 camadas idênticas. Além disso, inclui ambas as subcamadas já apresentadas no codificador, bem como as conexões residuais e a camada de normalização. Uma terceira subcamada que realiza atenção entre a saída do codificador e a entrada do decodificador é adicionada ao decodificador.

Essa subcamada de autoatenção na pilha do decodificador foi modificada para evitar que as posições atendam às posições subsequentes. Com a combinação do mascaramento adicionado ao fato dos *embeddings* de saída serem deslocados por uma posição, as predições para a posição i irão depender apenas das saídas desconhecidas das posições inferiores a i.

Mecanismo de atenção

Uma função de atenção é um mecanismo que associa uma consulta a um conjunto de pares chave-valor para gerar uma saída. Nesta abordagem, a consulta, as chaves e os valores são todos vetores. A saída é calculada como uma combinação ponderada dos valores, onde os pesos são definidos com base na compatibilidade entre a consulta e cada chave. A Figura 2.2 apresenta a estrutura de ambas a atenção por produto escalar escalonado e o mecanismo de autoatenção com vários cabeçotes.

Atenção por Produto Escalar Escalonado (Scaled Dot-Product Attention):

O modelo recebe três conjuntos de vetores: consulta $\mathbf{Q} \in \mathbb{R}^{d_k}$, chave $\mathbf{K} \in \mathbb{R}^{d_k}$, e valor $\mathbf{V} \in \mathbb{R}^{d_v}$. Cada um desses vetores é uma representação de informações diferentes, onde as consultas representam a informação que se deseja buscar, as chaves contêm a informação disponível para comparação, e os valores contêm a informação que será retornada com base na consulta.

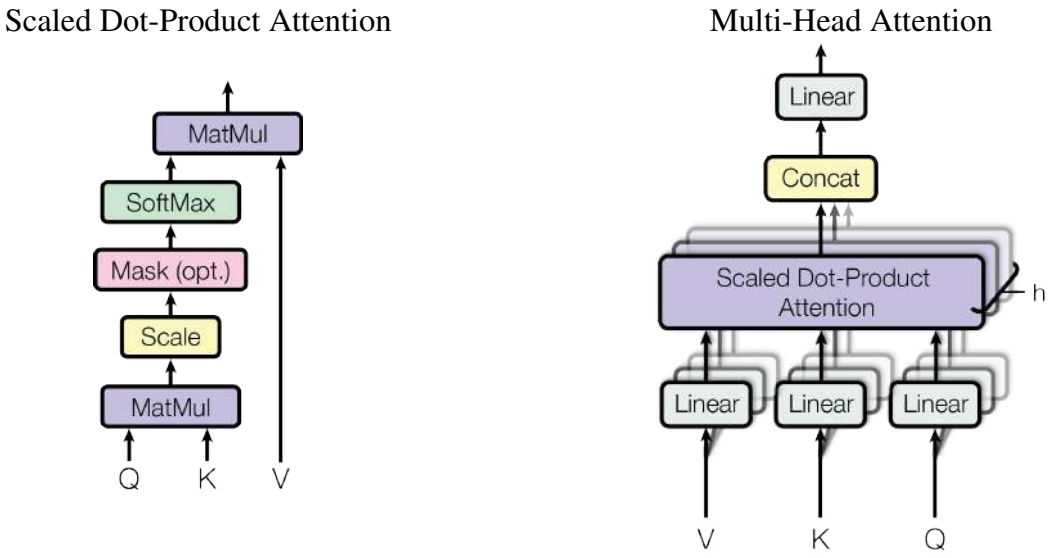


Figura 2.2: Atenção por produto escalar escalonado (Scaled Dot-Product Attention) à esquerda. Mecanismo de autoatenção com vários cabeçotes (Multi-Head Attention) à direita

Fonte: Vaswani et al. (2023)

Os pesos dos valores são obtidos através da aplicação da função *softmax* na divisão entre o produto escalar da consulta com todas as chaves e $\sqrt{d_k}$. A matriz de saída é dada por:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

A função de atenção proposta em Vaswani et al. (2023) é idêntica à atenção por produto escalar, porém com o acréscimo do fator de escalonamento de $1/\sqrt{d_k}$. Esse fator é adicionado à função para evitar problemas nos produtos escalares que tendem a ser grandes, o que faz com que a função *softmax* se mova para regiões onde possui gradientes muito pequenos. Esse problema ocorre com valores grandes de d_k.

Mecanismo de autoatenção com vários cabeçotes (Multi-Head Attention):

As entradas, representadas pelos vetores de consulta $\mathbf{Q} \in \mathbb{R}^{d_k}$, chave $\mathbf{K} \in \mathbb{R}^{d_k}$, e valor $\mathbf{V} \in \mathbb{R}^{d_v}$, são projetadas linearmente h vezes, resultando em múltiplas projeções para cada um desses vetores. A função de atenção é então aplicada em paralelo a cada um desses subespaços, produzindo um total de h cabeças de atenção, cada uma com dimensão d_v. As saídas de atenção dessas cabeças são posteriormente combinadas em um único vetor e projetadas novamente, gerando os valores finais. Cada cabeça de atenção $\mathbf{h}_i (i = 1, \ldots, h)$ é calculada como

$$\mathbf{h}_i = f\left(\mathbf{W}_i^{(Q)}\mathbf{Q}, \mathbf{W}_i^{(K)}\mathbf{K}, \mathbf{W}_i^{(V)}\mathbf{V}\right) \in \mathbb{R}^{d_v},$$

em que $\mathbf{W}_i^{(Q)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^{(K)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$ e $\mathbf{W}_i^{(V)} \in \mathbb{R}^{d_{\text{model}} \times d_v}$ são as matrizes de projeção para a i-ésima cabeça de atenção, e f representa a função de atenção escalonada (*scaled dot-product attention*).

Por fim, as saídas das cabeças de atenção são concatenadas e projetadas através de uma matriz de pesos final $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ para produzir a saída final do mecanismo de atenção com vários cabeçotes (Figura 2.2):

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}\left(\mathbf{h}_1, \ldots, \mathbf{h}_h\right) \mathbf{W}^O$$

No artigo original dos *Transformers*, Vaswani et al. (2023) utilizaram $h = 8$ camadas de atenção paralelas. Além disso, $d_k = d_v = d_{\text{model}}/h = 64$.

Na Figura 2.1 é possível observar que o mecanismo de atenção é utilizado de três jeitos distintos:

1. Nas camadas de "atenção entre codificador e decodificador", as consultas são derivadas da camada anterior do decodificador, e as chaves e valores de memória provêm da saída do codificador. Desse modo, cada posição no decodificador considera todas as posições na sequência de entrada, replicando os mecanismos tradicionais de atenção em modelos de sequência-para-sequência;
2. O codificador inclui camadas de autoatenção, em que as consultas, chaves e valores são todos extraídos da saída da camada anterior do codificador. Dessa forma, cada posição no codificador é capaz de atender a todas as posições na camada anterior;
3. De modo semelhante, as camadas de autoatenção no decodificador permitem que cada posição atenda a todas as posições anteriores até a posição atual. Todos os valores de entrada da função *softmax* que corresponderiam a conexões não permitidas são mascarados (definindo como $-\infty$) dentro da atenção por produto escalar escalonado, de forma a manter a propriedade auto-regressiva do modelo (ver Figura 2.2).

Redes Feed-Forward Posicionais (Position-wise Feed-Forward Networks):

Ambos o codificador e decodificador possuem em cada camada uma rede neural direta simples, posicional e totalmente conectada composta por duas funções de ativação *Rectified Linear Unit* (ReLU),

$$\text{FEN(x)} = \max(0, xW_1 + b_1)W_2 + b_2,$$

aplicadas separadamente e identicamente em cada posição, em que W_1 e W_2 são matrizes de pesos e b_1 e b_2 são vetores de viés (parâmetros treináveis). No *Transformer* original, a dimensão interna é $d_{ff} = 2048$, enquanto a entrada e a saída têm dimensão $d_{\text{model}} = 512$.

Embeddings e Softmax:

Assim como em outros modelos de transdução de sequência, as *embeddings* aprendidas foram utilizadas para transformar os *tokens* de entrada e saída em vetores de mesma dimensão do modelo (d_{model}). Além disso, ambas a transformação linear e a função *softmax* são utilizadas no final do processo de decodificação para converter o vetor de saída em probabilidades previstas para o próximo *token*. Por fim, a mesma matriz de pesos foi compartilhada entre as duas camadas de *embedding* e a transformação linear pré-softmax, com uma multiplicação adicional desses pesos por $\sqrt{d_{\text{model}}}$ nas camadas de *embedding*.

Codificação posicional (Positional Encoding):

Diferentemente de abordagens sequenciais tradicionais, como as redes recorrentes e de convolução, o modelo *Transformer* não processa a ordem das palavras, então a informação sobre a posição relativa ou absoluta dos *tokens* nas sequências de entrada precisa ser inserida de alguma forma. Para lidar com esse problema, Vaswani et al. (2023) propõem a adição de informações de posicionamento aos *embeddings* de entrada na parte inferior das pilhas do codificador e do decodificador.

A codificação posicional é um conjunto de vetores baseado nas funções seno e cosseno, projetado de forma a fornecer ao modelo informações sobre a posição relativa das palavras na sequência, podendo assim levar em consideração a ordem das palavras sem depender de uma estrutura sequencial fixa. A codificação posicional tem a mesma dimensão dos *embeddings*, de modo que os dois possam ser somados.

Existem diversas escolhas de codificação posicional, embora a proposta do modelo *Transformer* tenha sido definida de maneira sinusoidal. Essa escolha foi estabelecida devido à hipótese dos autores de que essa função iria permitir que o modelo facilmente "prestasse atenção" nas posições relativas. Além disso, a escolha do modelo sinusoide deu-se pelo fato de permitir que o modelo extrapole para tamanhos de sequências maiores do que os encontrados no treinamento. As funções seno e cosseno são definidas por:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\mathrm{model}}}\right) \qquad \text{e} \qquad \mathrm{PE}_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\mathrm{model}}}\right),$$

onde pos é a posição, i é a dimensão e d_{model} é a dimensão dos *embeddings*.

A partir da Figura 2.1 é possível observar que a arquitetura geral dos *transformers* utiliza, tanto no codificador quanto no decodificador, o mecanismo de autoatenção e a rede neural totalmente conectada.

2.2.2.2 Bidirectional Encoder Representations from Transformers (BERT)

O BERT (Representações de Codificador Bidirecional de Transformadores, do inglês *Bidirectional Encoder Representations from Transformers*) é um modelo de representação de linguagem desenvolvido por Devlin et al. (2019), que utiliza a arquitetura *Transformers* (Vaswani et al., 2017) para pré-treinar representações bidirecionais ao condicionar conjuntamente contexto à esquerda e á direita em todas as camadas.

A estrutura do BERT foi desenvolvida em duas etapas: pré-treinamento e ajuste fino. No pré-treinamento, o modelo é treinado com dados não rotulados em diversas tarefas específicas, enquanto no ajuste fino ele é inicializado com os parâmetros pré-treinados, que são então ajustados usando dados rotulados das tarefas subsequentes. Embora essas tarefas sejam inicializadas com os mesmos parâmetros pré-treinados, cada uma possui um modelo ajustado separadamente. A Figura 2.3 abaixo é um exemplo de uma tarefa de pergunta-resposta.

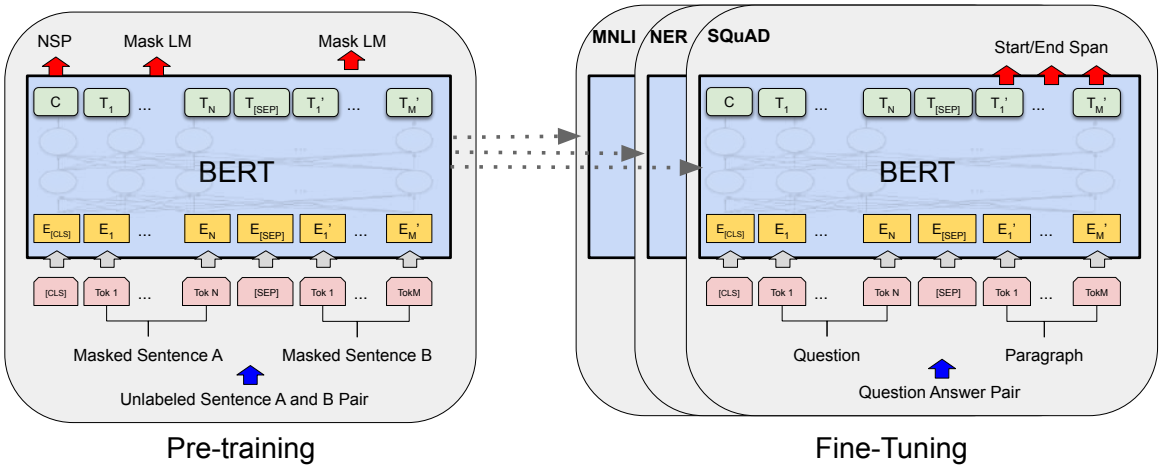


Figura 2.3: Procedimentos gerais de pré-treinamento e ajuste fino para o BERT. Exceto pelas camadas de saída, as mesmas arquiteturas são usadas tanto no pré-treinamento quanto no ajuste fino.
Fonte: Devlin et al. (2019)

Arquitetura do BERT

A arquitetura do modelo BERT é um codificador *Transformer* bidirecional de múltiplas camadas baseado em Vaswani et al. (2017). Considere L como o número de camadas, H como o tamanho da camada oculta e A o número de cabeçotes de autoatenção. No artigo de introdução do BERT, os autores apresentaram os resultados para dois modelos distintos: $\textbf{BERT}_{\textbf{BASE}}$ e $\textbf{BERT}_{\textbf{LARGE}}$, com 110 milhões ($L = 12$, $H = 768$, $A = 12$) e 340 milhões (com $L = 24$, $H = 1024$, $A = 16$) de parâmetros, respectivamente (Devlin et al., 2019).

Representações de entrada/saída

A representação de entrada, chamada de"sequência", é capaz de representar tanto uma única frase quanto um par de frases em uma sequência de *token*, permitindo que o modelo lide com uma variedade de tarefas. Na estrutura do modelo, o primeiro *token* é sempre um *token* de classificação especial, denominado [CLS]. As sequências são diferenciadas por meio de um *token* especial ([SEP]) e pela adição de um *embedding* aprendido a cada *token*, indicando a qual frase ele pertence.

O BERT utiliza *embeddings* `WordPiece` (Wu et al., 2016), que segmentam palavras em subunidades menores, priorizando radicais e, em seguida, adicionando sufixos ou prefixos. Essa técnica permite que o modelo represente palavras de forma mais flexível, decompondo-as em partes conhecidas.

Como apresentado na Figura 2.3, os *embeddings* de entrada são detonados como E, o vetor final oculto do *token* especial [CLS] de $C \in \mathbb{R}^H$ e o vetor final oculto para o $i-$ésimo *token* de entrada como $T_i \in \mathbb{R}^H$. Para um dado *token*, sua representação de entrada é construída através da soma dos *embedding* correspondentes de *token*, segmento e posição, conforme Figura 2.4.

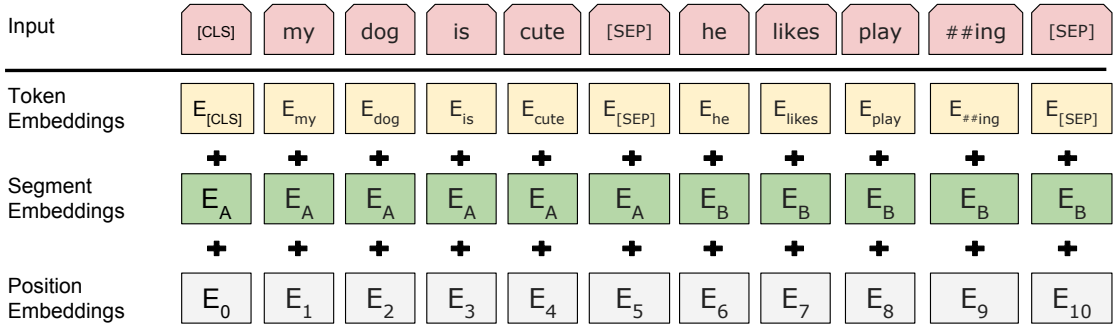


Figura 2.4: Representação de entrada do BERT.
Fonte: Devlin et al. (2019)

Pré-treinamento

De acordo com Devlin et al. (2019), o BERT é pré-treinado utilizando duas tarefas não-supervisionadas: modelagem de linguagem mascarada (do inglês *Masked Language Modeling*, MLM) e predição da próxima frase (do inglês *Next Sentence Prediction*, NSP).

Na **Modelagem de Linguagem Mascarada**, o treinamento do modelo profundo bidirecional é realizado ao mascarar aleatoriamente uma porcentagem dos *tokens* de entrada e, em seguida, prever esses *tokens* mascarados com base no contexto dos outros *tokens* da frase. Esse procedimento também é conhecido na literatura como *"cloze task"* (Taylor, 1953).

Embora a técnica seja eficaz para o treinamento bidirecional, ela apresenta algumas limitações. Como o token `[MASK]` aparece apenas durante o pré-treinamento e não no ajuste fino, isso gera uma discrepância entre as duas etapas. Para contornar esse problema, Devlin et al. (2019) decidiram não substituir sempre a palavra mascarada pelo `[MASK]`. Nos experimentos, 15% de todos os *tokens* do tipo *WordPiece* são escolhidos aleatoriamente para mascaramento. O processo ocorre da seguinte forma: quando o i-ésimo *token* é selecionado, ele é substituído pelo *token* `[MASK]` em 80% das vezes, por um *token* aleatório em 10% das vezes e pelo próprio *token* original em 10% das vezes. Após, T_i é utilizado para prever o *token* original, com a perda sendo calculada pela entropia cruzada.

A **Previsão da Próxima Sentença** tem como objetivo treinar um modelo que compreenda as relações entre sentenças. Muitas tarefas de NLP, como *Question Answering* (QA) e *Natural Language Inference* (NLI), exigem que o modelo consiga identificar se há uma conexão lógica ou semântica entre duas sentenças. O NSP prepara o BERT para esse tipo de análise.

Para que o modelo aprenda a compreender relações entre sentenças, Devlin et al. (2019) utilizaram uma tarefa binária de previsão da próxima sentença durante o pré-treinamento. Nesse processo, são selecionadas duas sentenças, A e B. Em 50% dos casos, B é a continuação real de A (rotulada como `IsNext`), enquanto que nos outros 50%, B é uma sentença aleatória retirada do *corpus* (rotulada como `NotNext`). Na Figura 2.3, o contexto `C` é utilizado para realizar a previsão da próxima sentença.

Ajuste fino

O mecanismo de autoatenção do BERT simplifica o ajuste fino, permitindo que o modelo seja facilmente adaptado a diferentes tipos de tarefas. Para isso, é necessário apenas configurar as entradas e saídas específicas de cada tarefa e ajustar todos os seus parâmetros de maneira integrada.

Na etapa de entrada, as sentenças A e B utilizadas no pré-treinamento são associadas a distintos cenários: pares de sentenças para tarefas de parafraseamento, pares hipótese-premissa para implicação textual, pares pergunta-trecho em QA e pares texto-$\emptyset$ para tarefas como classificação de texto ou rotulação de sequência. Na etapa de saída, as representações dos *tokens* são direcionadas para uma camada de saída em tarefas que operam no nível de *token*, como rotulação de sequência ou QA. Já a representação `[CLS]` é destinada a uma camada de saída voltada para tarefas de classificação, como análise de sentimento ou implicação textual (Devlin et al., 2019).

2.3 Redução de dimensionalidade

Apesar do espaço vetorial dos *embeddings* das palavras ser k-dimensional, representações bi-dimensionais são utilizadas para visualizar as representações de forma mais simplificada (Lane et al., 2019). Desta forma, redução de dimensionalidade é frequentemente aplicada para projetar os *embeddings* de alta dimensão em um espaço de duas dimensões. Técnicas como Análise de Componentes Principais (PCA) ou *t-Distributed Stochastic Neighbor Embedding* (t-SNE) são comumente usadas para esse propósito (Liu et al., 2018), pois permitem uma visualização mais intuitiva das relações semânticas entre as palavras, facilitando a interpretação e identificação de padrões, mesmo em espaços de menor dimensionalidade.

Além do PCA e do t-SNE, a técnica *Uniform Manifold Approximation and Projection* (UMAP) também é amplamente utilizada para esse propósito. O UMAP tem ganhado popularidade por sua capacidade de preservar tanto a estrutura local quanto a global dos dados em alta dimensionalidade, facilitando uma visualização mais clara das relações entre as representações semânticas das palavras (McInnes et al., 2018).

Uniform Manifold Approximation and Projection (UMAP)

O UMAP (*Uniform Manifold Approximation and Projection*), proposto por McInnes et al. (2018), baseia-se em fundamentos da teoria das variedades e da análise topológica de dados. O método parte da construção de um grafo de k-vizinhos mais próximos no espaço original e define, para cada ponto x_i, uma conectividade difusa com seus vizinhos x_j por meio de pesos $w_{ij} \in (0, 1)$. Em uma formulação padrão, esses pesos podem ser definidos por

$$w_{ij} = \exp\left(-\frac{\max\{0,\, d(x_i, x_j) - \rho_i\}}{\sigma_i}\right),$$

em que $d(\cdot, \cdot)$ é a métrica no espaço original, ρ_i é um ajuste local (associado à distância ao vizinho mais próximo) e σ_i controla a escala local da vizinhança (McInnes et al., 2018). Para obter uma conectividade simétrica, combina-se a conectividade nas duas direções, por exemplo por uma união

difusa do tipo

$$w_{ij}^{(sym)} = w_{ij} + w_{ji} - w_{ij}w_{ji}.$$

A partir dessa representação difusa no espaço original, o UMAP busca uma projeção em baixa dimensão y_i cuja estrutura de vizinhança seja equivalente. Para isso, define-se uma conectividade no espaço reduzido w'_{ij} como uma função decrescente da distância euclidiana $\|y_i - y_j\|$, frequentemente escrita como

$$w'_{ij} = \frac{1}{1 + a\|y_i - y_j\|^{2b}},$$

em que a e b determinam o formato do decaimento e se relacionam aos hiperparâmetros que controlam o grau de compactação no mapa (McInnes et al., 2018). Por fim, o *layout* em baixa dimensão é obtido por otimização de uma função objetivo baseada em entropia cruzada entre as conectividades do espaço original e do espaço reduzido, que pode ser expressa como

$$C = \sum_{i \neq j} \left[w_{ij} \log\left(\frac{w_{ij}}{w'_{ij}}\right) + (1 - w_{ij}) \log\left(\frac{1 - w_{ij}}{1 - w'_{ij}}\right) \right].$$

Dessa forma, o método penaliza situações em que pontos fortemente conectados no espaço original se afastam no espaço reduzido e, simultaneamente, evita que pontos sem conectividade relevante sejam indevidamente aproximados, preservando relações locais e conectividade topológica (McInnes et al., 2018).

2.4 Similaridade entre palavras

Uma das abordagens amplamente reconhecidas para calcular a similaridade entre vetores é através da distância euclidiana. No entanto, ao lidarmos com palavras, é necessário que o método utilizado preserve o sentido geral do texto (Pal et al., 2022). Ao trabalharmos com NLP e realizarmos uma redução de dimensionalidade, utilizar o ângulo entre dois vetores é o método mais popular.

Similaridade de Cosseno

A similaridade de cossenos é amplamente utilizada no domínio de classificação de texto devido à sua eficiência computacional e bom desempenho (Li and Han, 2013; Park et al., 2020; Sak, 2023).

Considere dois vetores $\vec{v}$ e $\vec{w}$ n-dimensional. A similaridade de cossenos $\cos(\vec{v}, \vec{w})$ é representada pelo produto escalar dos dois vetores normalizados, ou seja

$$\text{similaridade de cosseno } := \cos(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}, \tag{2.6}$$

em que v_i e w_i são os i-ésimos componentes dos vetores $\vec{v}$ e $\vec{w}$, respectivamente. A similaridade de cosseno sempre pertence ao intervalo $[-1, 1]$ (Li and Han, 2013). Como o cosseno tem a propriedade de ser 1 para vetores idênticos (que apontam para a mesma direção) e 0 para vetores ortogonais, o cosseno do ângulo entre os dois vetores é utilizado como uma similaridade numérica (Singhal, 2001).

2.5 Agrupamento de palavras

O agrupamento de palavras no contexto de NLP, especialmente com o uso de *embeddings* contextuais, visa capturar a semelhança semântica entre palavras em grandes *corpora* textuais. *Embeddings* como os do BERT (Devlin et al., 2019) representam palavras de forma dinâmica, adaptando-se ao contexto em que ocorrem, o que torna os métodos tradicionais de agrupamento desafiadores, pois as palavras podem ter significados diferentes dependendo da situação. Nesse contexto, as abordagens eficazes são aquelas que consideram dados de alta dimensionalidade e estruturas complexas, em que não se pode assumir formas ou densidades uniformes nos grupos.

Hierarchical Density-Based Clustering (HDBSCAN)

O algoritmo HDBSCAN (*Hierarchical Density-Based Spatial Clustering of Applications with Noise*) é uma técnica de agrupamento baseada em densidade que estende o algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) ao incorporar uma abordagem hierárquica para identificar agrupamentos em conjuntos de dados com densidades variadas, sem a necessidade de definir previamente um parâmetro de distância global, ou seja, um raio de vizinhança fixo (Campello et al., 2013).

O algoritmo utiliza uma distância de alcançabilidade mútua para construir uma hierarquia de *clusters* baseada em densidade. A partir dessa árvore hierárquica de conectividade, os agrupamentos mais estáveis são extraídos com base em um critério de persistência. A distância de alcançabilidade mútua entre dois pontos a e b é apresentada na Definição 7 do artigo de Campello et al. (2013) como:

$$d_{\text{alcançabilidade}}(a, b) = \max\{d_{\text{core}}(a), d_{\text{core}}(b), d(a, b)\},$$

em que $d_{\text{core}}(x)$ representa a distância do ponto x ao seu k-ésimo vizinho mais próximo e $d(a, b)$ é a distância entre a e b segundo uma métrica apropriada, como a Euclidiana. Esse valor reflete a densidade local ao redor dos pontos e penaliza conexões em regiões menos densas, ou seja, assegura que regiões de menor densidade sejam menos propensas a formar *clusters*, promovendo a robustez do algoritmo frente a variações de densidade.

Essa métrica é então utilizada na construção de um grafo de vizinhança ponderado, cujas arestas são ponderadas por essa distância, a partir do qual se extrai uma árvore de expansão mínima (MST, do inglês *Minimum Spanning Tree*). A MST é então utilizada para criar uma hierarquia de *clusters* por meio da eliminação progressiva de arestas com maiores pesos, correspondendo a regiões de menor densidade. Essa hierarquia reflete a estrutura dos dados em diferentes níveis de densidade.

A noção de estabilidade como critério central para a seleção de *clusters* mais significativos da hierarquia de densidade gerada pelo HDBSCAN parte do princípio que *clusters* com alta estabilidade são aqueles que permanecem densamente conectados ao longo de uma ampla gama de valores de densidades, sendo assim, mais robustos. Esse conceito é formalizado a partir da adaptação do conceito de "excesso de massa" (Müller and Sawitzki, 1991) para o conceito de "excesso de massa relativa" (*Relative Excess Mass*) de um *cluster* $\mathbf{C}_i$ (Campello et al., 2013).

Para um conjunto de dados finito $\mathbf{X}$, com hierarquia HDBSCAN construída a partir de rótulos de

clusters e níveis de densidade discretos λ, para cada *cluster* $\mathbf{C}_i$ a estabilidade pode ser expressa como:

$$S(\mathbf{C}_i) = \sum_{x_j \in \mathbf{C}_i} (\lambda_{max}(\mathbf{x}_j, \mathbf{C}_i) - \lambda_{min}(\mathbf{C}_i))$$

em que $\lambda_{min}(\mathbf{C}_i)$ é o nível mínimo de densidade no qual $\mathbf{C}_i$ existe e $\lambda_{max}(\mathbf{x}_j, \mathbf{C}_i)$ é o nível de densidade além do qual o objeto x_j deixa de pertencer ao *cluster* $\mathbf{C}_i$. *Clusters* com maior estabilidade são selecionados como representações finais, e os pontos que não permanecem em nenhum *cluster* estável são rotulados como ruído.

Capítulo 3

Grafos

Neste capítulo, é apresentada inicialmente uma revisão sobre **Teoria dos Grafos**, abordando seus conceitos fundamentais e formalizações matemáticas, incluindo a definição de grafos, vértices e arestas, bem como as principais representações matriciais utilizadas para descrever a estrutura de redes. Em seguida, são explorados aspectos relacionados à teoria espectral de grafos, com ênfase no **Laplaciano do grafo** e em suas propriedades, que constituem a base para a caracterização estrutural de redes complexas. Especificamente, são discutidas abordagens voltadas à **comparação estrutural entre grafos**, destacando o uso de métodos baseados em propriedades espectrais para a análise de similaridade entre redes de diferentes tamanhos. Nesse contexto, é apresentado o descritor espectral **NetLSD** (*Network Laplacian Spectral Descriptor*), selecionado como o método central desta dissertação para a comparação das redes de interação entre jogadores.

3.1 Definições básicas

Os grafos constituem estruturas matemáticas utilizadas para modelar sistemas compostos por unidades que interagem entre si. Formalmente, um grafo é definido como um par

$$G = (V, E),$$

em que $V = \{v_1, \ldots, v_n\}$ é um conjunto finito e não vazio chamado conjunto de vértices e $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$ é um conjunto de pares não ordenados de vértices distintos, chamado conjunto de arestas (West, 1996; Diestel, 2017). Cada vértice corresponde a uma entidade do sistema analisado, enquanto as arestas representam relações ou interações entre dois vértices.

A ordem de um grafo G é definida como o número de vértices, representado por $|V|$, enquanto o número de arestas é denotado por $|E|$. A partir da ordem, os grafos podem ser caracterizados como finitos, infinitos ou enumeráveis (Diestel, 2017).

Em grafos simples e não direcionados, uma aresta E é definida como um par não ordenado $\{u, v\}$. Além disso, não existem laços, ou seja, $\{v, v\} \notin E$ e não há arestas múltiplas entre o mesmo par de vértices. Em contrapartida, em grafos direcionados o conjunto de arestas é um subconjunto de $V \times V$, isto é, $E \subseteq \{(u, v) : u, v \in V\}$, e de modo geral que $(u, v) \neq (v, u)$, caracterizando orientação na relação (West, 1996).

Grafos podem ainda ser ponderados ou não ponderados. Em grafos ponderados, define-se uma função peso

$$w : E \to \mathbb{R},$$

em que $w(u, v)$ expressa, por exemplo, intensidade, proximidade, frequência ou custo da interação entre u e v (Newman, 2010). Além disso, a estrutura de um grafo G com $n = |V|$ vértices pode ser representada por meio da matriz de adjacência $\mathbf{A} = (a_{ij})_{n\times n}$, definida por

$$a_{ij} = \begin{cases} w(v_i, v_j), & \text{se } (v_i, v_j) \in E, \\ 0, & \text{caso contrário,} \end{cases} \tag{3.1}$$

em grafos ponderados, e por $a_{ij} \in \{0, 1\}$ em grafos não ponderados (West, 1996; Newman, 2010).

A partir dessa representação matricial, torna-se possível investigar propriedades estruturais de sistemas complexos por meio de conceitos fundamentais da teoria dos grafos. Conceitos como grau do vértice, caminho, distância e conectividade são fundamentais nesse contexto.

O grau de um vértice $v \in V$ em um grafo G é definido como a soma dos pesos das arestas incidentes a ele, isto é,

$$deg(v) = \sum_{i=1}^{n} w(v, v_i). \tag{3.2}$$

E quando o grafo G for não ponderado, esse número coincide com o número de arestas incidentes a ele, isto é,

$$deg(v) = |\{u \in V : \{u, v\} \in E\}|. \tag{3.3}$$

Um caminho entre os vértices u e v é uma sequência finita de vértices distintos $(v_0, \ldots, v_k)$ tal que $v_0 = u$ e $v_k = v$ e $\{v_{i-1}, v_i\} \in E$ para todo $i = 1, \ldots, k$. Se $v_0 = v_k$ então temos um caminho fechado, denominado ciclo. Além disso, distância $d(u, v)$ é definida como o comprimento do menor caminho entre u e v. Por fim, um grafo é conexo quando $d(u, v) < \infty$ para quaisquer $u, v \in V$ (Diestel, 2017).

A partir dessas definições básicas, pode-se descrever tanto propriedades locais quanto globais de um grafo. Em nível local, quantidades como o grau dos vértices e a distribuição de comprimentos de caminhos capturam a estrutura imediata de vizinhança. Em nível global, distâncias entre pares de vértices e a conectividade do grafo permitem caracterizar o alcance e a coesão da rede (Diestel, 2017; Newman, 2010).

Em particular, o grau de um vértice descreve o número de adjacências diretas, enquanto caminhos e distâncias estão associados à possibilidade e ao custo de deslocamento entre vértices ao longo da estrutura. Esses conceitos constituem a base para métricas de centralidade e para a análise de redes observadas em sistemas reais, nas quais propriedades estruturais traduzem padrões de interação em domínios como redes sociais, biológicas, tecnológicas e esportivas. (Diestel, 2017; Newman, 2010).

3.2 Representações matriciais e Laplaciano do grafo

Conforme apresentado na Equação 3.1, a estrutura de um grafo pode ser convenientemente expressa por meio de uma representação matricial chamada matriz de adjacência **A**. Além dessa matriz, define-se a matriz de grau **D**, uma matriz diagonal cujas entradas correspondem ao grau de cada vértice, conforme Equação 3.3. Com base nessas duas matrizes, obtém-se o Laplaciano do grafo,

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{3.4}$$

o qual desempenha papel central na teoria espectral de grafos (Chung, 1997). Além disso, o Laplaciano do grafo aparece em diferentes contextos teóricos e aplicados, incluindo modelos de caminhadas aleatórias em redes, particionamento de grafos e estudos de conectividade em redes (Newman, 2010).

O Laplaciano **L** é uma matriz simétrica cujos autovalores são reais e não negativos (Newman, 2010). O espectro dessa matriz contém informações relevantes sobre a estrutura do grafo. Em particular, a multiplicidade do autovalor zero é igual ao número de componentes conexas, estabelecendo uma relação direta entre propriedades espectrais e topológicas da rede (Chung, 1997).

Uma variante amplamente utilizada dessa construção é o Laplaciano normalizado, bastante adotado em aplicações espectrais, definido como

$$\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2},$$

o qual busca atenuar o efeito das variações no grau dos vértices sobre as propriedades espectrais do grafo.

Essa normalização torna a representação mais estável em redes com heterogeneidade de conectividade, permitindo comparações mais consistentes entre grafos com diferentes distribuições de grau (Chung, 1997). Em função dessas propriedades, o Laplaciano normalizado é amplamente empregado em processos de difusão em redes, particionamento espectral e em métodos baseados em descritores espectrais para a análise estrutural de grafos complexos (Newman, 2010).

Sob essa perspectiva, descritores espectrais fundamentados no Laplaciano assumem um papel central na caracterização estrutural e na comparação entre grafos, uma vez que sintetizam propriedades topológicas globais a partir do espectro da matriz Laplaciana (Chung, 1997; Newman, 2010).

3.3 Comparação estrutural entre grafos

A comparação entre grafos é uma tarefa central na análise de redes. Entretanto, diferentes grafos podem apresentar tamanhos distintos, distribuições de grau heterogêneas e padrões topológicos variados, o que dificulta a definição de medidas de similaridade baseadas apenas em estatísticas locais ou propriedades pontuais da rede (Newman, 2010).

Entre as abordagens existentes, métodos baseados em propriedades espectrais ocupam posição de destaque, pois permitem representar grafos por meio de vetores compactos capazes de capturar características estruturais globais, viabilizando a comparação direta entre diferentes redes (Newman,

2010). Ao utilizar informações espectrais, torna-se possível comparar grafos independentemente do número de vértices, o que é particularmente relevante em cenários nos quais as redes apresentam tamanhos distintos (Tsitsulin et al., 2018).

Dessa forma, métodos baseados no espectro do Laplaciano fornecem uma representação estrutural robusta, permitindo a análise de similaridade entre grafos e a identificação de padrões estruturais compartilhados entre diferentes redes (Chung, 1997; Newman, 2010).

Apesar dessas vantagens, a utilização direta do espectro do Laplaciano apresenta limitações práticas, uma vez que diferentes grafos podem possuir espectros de tamanhos distintos e elevada sensibilidade a variações locais da estrutura. Em função disso, torna-se necessário o uso de **descritores espectrais** capazes de resumir essas informações em representações vetoriais estáveis e comparáveis entre diferentes redes (Tsitsulin et al., 2018).

3.3.1 Descritor espectral NetLSD

Com base nesse princípio, Tsitsulin et al. (2018) propuseram o NetLSD (*Network Laplacian Spectral Descriptor*), um descritor espectral projetado para representar grafos por meio de uma assinatura vetorial derivada do espectro do Laplaciano. O objetivo do método é capturar propriedades estruturais globais da rede, permitindo a comparação entre grafos com diferentes tamanhos e densidades.

O NetLSD é fundamentado na chamada *heat trace*, definida a partir dos autovalores do Laplaciano normalizado. Essa abordagem modela o comportamento de um processo de difusão sobre o grafo ao longo do tempo, possibilitando a extração de informações estruturais em múltiplas escalas temporais (Tsitsulin et al., 2018). Em particular, tempos menores estão associados a propriedades locais da rede, enquanto tempos maiores capturam padrões globais de conectividade.

Formalmente, seja $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ o conjunto de autovalores do Laplaciano normalizado de um grafo com n vértices. A *heat trace* é definida como

$$h(t) = \sum_{i=1}^{n} e^{-t\lambda_i}, \tag{3.5}$$

em que t representa o parâmetro temporal do processo de difusão. A assinatura do NetLSD é obtida avaliando-se essa função para um conjunto discreto de valores de t, resultando em um vetor que sintetiza o comportamento espectral do grafo ao longo de diferentes escalas (Tsitsulin et al., 2018).

Uma das principais vantagens do NetLSD é sua invariância ao tamanho do grafo, o que permite a comparação direta entre redes com números distintos de vértices. Além disso, o método apresenta robustez a pequenas perturbações estruturais, característica relevante em aplicações envolvendo dados reais, nos quais ruídos e variações locais são frequentes (Tsitsulin et al., 2018).

No contexto desta dissertação, o NetLSD é utilizado para comparar estruturalmente as redes de interação entre jogadores dos clubes da Serie A italiana, permitindo identificar padrões globais de organização tática e similaridades estruturais entre os clubes ao longo da temporada analisada.

Capítulo 4

Metodologia

A metodologia proposta transforma dados tabulares de eventos de partidas de futebol em representações adequadas para análises baseadas em NLP e redes. A Figura 4.1 apresenta o panorama geral da metodologia.



Figura 4.1: Panorama geral da metodologia.

1. Inicialmente, a **base de dados** em formato `JSON` é importada para o ambiente `R` e submetida a etapas de **preparação e pré-processamento**, incluindo seleção e padronização de variáveis e tratamentos necessários para a construção do *corpus*.

2. Em seguida, adota-se uma abordagem de *data-to-text generation* (DTG) baseada em regras e *templates*, na qual cada evento é convertido em uma sentença padronizada e posteriormente normalizada, permitindo interpretar a partida como uma sequência textual.

3. Com o *corpus* construído, são extraídas **representações distribuídas** por dois modelos (GloVe e BERT). No caso do BERT, devido ao tamanho dos textos, emprega-se **segmentação com sobreposição** para preservar continuidade contextual, e os *embeddings* contextualizados são armazenados para análises subsequentes,

4. incluindo uma etapa exploratória de inspeção do espaço vetorial com **UMAP e HDBSCAN**.

5. No componente relacional, para cada equipe é construída uma **rede não direcionada e ponderada** em que os nós representam jogadores e as arestas são definidas pela similaridade de cosseno entre vetores associados aos atletas, com aplicação de um deslocamento aditivo para garantir pesos não negativos; para reduzir densidade e favorecer interpretação, aplica-se um *backbone* adaptativo por percentil.

6. As redes são então analisadas por **métricas de centralidade** e comparadas globalmente via **NetLSD**, definindo-se distâncias entre assinaturas espectrais, o que permite realizar agrupamento hierárquico e investigar padrões estruturais entre equipes, com apoio do índice de *silhouette* na escolha do número de grupos.

4.1 Base de dados

O conjunto de dados utilizado neste estudo foi coletado e disponibilizado pela `Wyscout` (https://apidocs.wyscout.com/), uma empresa de referência em análises de futebol. O banco de dados contém informações detalhadas sobre eventos de partidas provenientes de diferentes competições de destaque, abrangendo exclusivamente a temporada **2017/18** das cinco principais ligas europeias — La Liga, Serie A, Premier League, Bundesliga e Ligue 1, além de incluir dados da Copa do Mundo FIFA de 2018 e do Campeonato Europeu da UEFA de 2016 (Pappalardo et al., 2019).

Neste estudo, o foco recai exclusivamente sobre a Serie A, a primeira divisão italiana. A temporada 2017/2018 é composta por 647.372 registros de eventos envolvendo 686 jogadores distribuídos em 380 partidas. Ao final da competição, a Juventus sagrou-se campeã, seguida por Napoli e Roma nas posições subsequentes da tabela, enquanto Benevento, Hellas Verona e Crotone ocuparam as últimas posições e foram rebaixados, conforme apresentado na Tabela 4.1. Uma descrição completa da estrutura e das variáveis do conjunto de dados pode ser encontrada em Pappalardo et al. (2019).

O conjunto de dados foi originalmente estruturado em formato tabular, em que cada linha representa um evento registrado durante uma partida e cada coluna corresponde a um atributo específico. Neste estudo, são consideradas informações referentes ao tipo de evento, aos jogadores envolvidos, às suas posições e equipes, ao local do evento (isto é, a posição no campo onde ocorreu) e, no caso de finalizações, ao desfecho da tentativa.

4.2 Preparação e pré-processamento dos dados

Segmentação do campo

Inicialmente, o conjunto de dados fornece as coordenadas (x, y) que representam a localização de cada evento em campo. Tanto x quanto y variam de 0 a 100 ao longo do eixo defesa–ataque, em que $x < 50$ indica que o evento ocorreu no campo defensivo da equipe. Para incorporar a dimensão espacial de forma interpretável, o campo foi dividido em 16 zonas distintas, abrangendo tanto a metade ofensiva quanto a defensiva. Essas zonas incluem áreas específicas como:

Tabela 4.1: Classificação final da Serie A italiana – Temporada 2017/2018

Posição	Clube	Jogos	Vitórias	Empates	Derrotas	Saldo de Gols	Pontos
1º	Juventus **(C)**	38	30	5	3	+62	95
2º	Napoli	38	28	7	3	+48	91
3º	Roma	38	23	8	7	+33	77
4º	Inter	38	20	12	6	+36	72
5º	Lazio	38	21	9	8	+40	72
6º	Milan	38	18	10	10	+14	64
7º	Atalanta	38	16	12	10	+18	60
8º	Fiorentina	38	16	9	13	+8	57
9º	Torino	38	13	15	10	+8	54
10º	Sampdoria	38	16	6	16	-4	54
11º	Sassuolo	38	11	10	17	-30	43
12º	Genoa	38	11	8	19	-10	41
13º	Chievo	38	10	10	18	-23	40
14º	Udinese	38	12	4	22	-15	40
15º	Bologna	38	11	6	21	-12	39
16º	Cagliari	38	11	6	21	-28	39
17º	SPAL	38	8	14	16	-20	38
18º	Crotone **(R)**	38	9	8	21	-26	35
19º	Hellas Verona **(R)**	38	7	4	27	-48	25
20º	Benevento **(R)**	38	6	3	29	-51	21

Regras para classificação: 1) Pontos; 2) Pontos no confronto direto; 3) Diferença de gols no confronto direto; 4) Diferença de gols; 5) Gols marcados; 6) Empate.
Abreviações: (C) Campeão; (R) Rebaixado.

- **Zona da Pequena Área (ZPA):** região imediatamente em frente ao gol, delimitada pela pequena área, onde ocorrem ações de alta pressão ofensiva e defensiva.
- **Zona da Grande Área (ZGA):** área ampliada em frente ao gol, incluindo a grande área, associada a finalizações, defesas e jogadas decisivas.
- **Zona Central Menor (ZCM):** região do círculo central, ponto de início das jogadas e transições.
- **Zona do Lado Esquerdo (ZLE):** faixa lateral esquerda do campo, voltada para ações ofensivas ou defensivas pelas extremidades.
- **Zona do Lado Direito (ZLD):** faixa lateral direita do campo, equivalente à ZLE, mas no lado oposto.
- **Zona Central (ZC):** área central ampliada, fundamental para construção de jogadas, marcação e controle de meio-campo.
- **Zona Central Esquerda (ZCE):** metade esquerda da zona central, geralmente associada à circulação de bola e cobertura defensiva no setor.
- **Zona Central Direita (ZCD):** metade direita da zona central, desempenhando funções equivalentes à ZCE, mas no setor oposto.

Ao longo das análises e na implementação computacional, as zonas foram rotuladas em inglês para manter consistência com a nomenclatura utilizada nos dados e nos resultados. A lista completa de zonas e seus respectivos rótulos encontra-se no Anexo A.

A Figura 4.2 apresenta a segmentação total do campo, conforme coordenadas originais do evento. A ideia central é capturar a localização espacial em que o evento ocorreu.

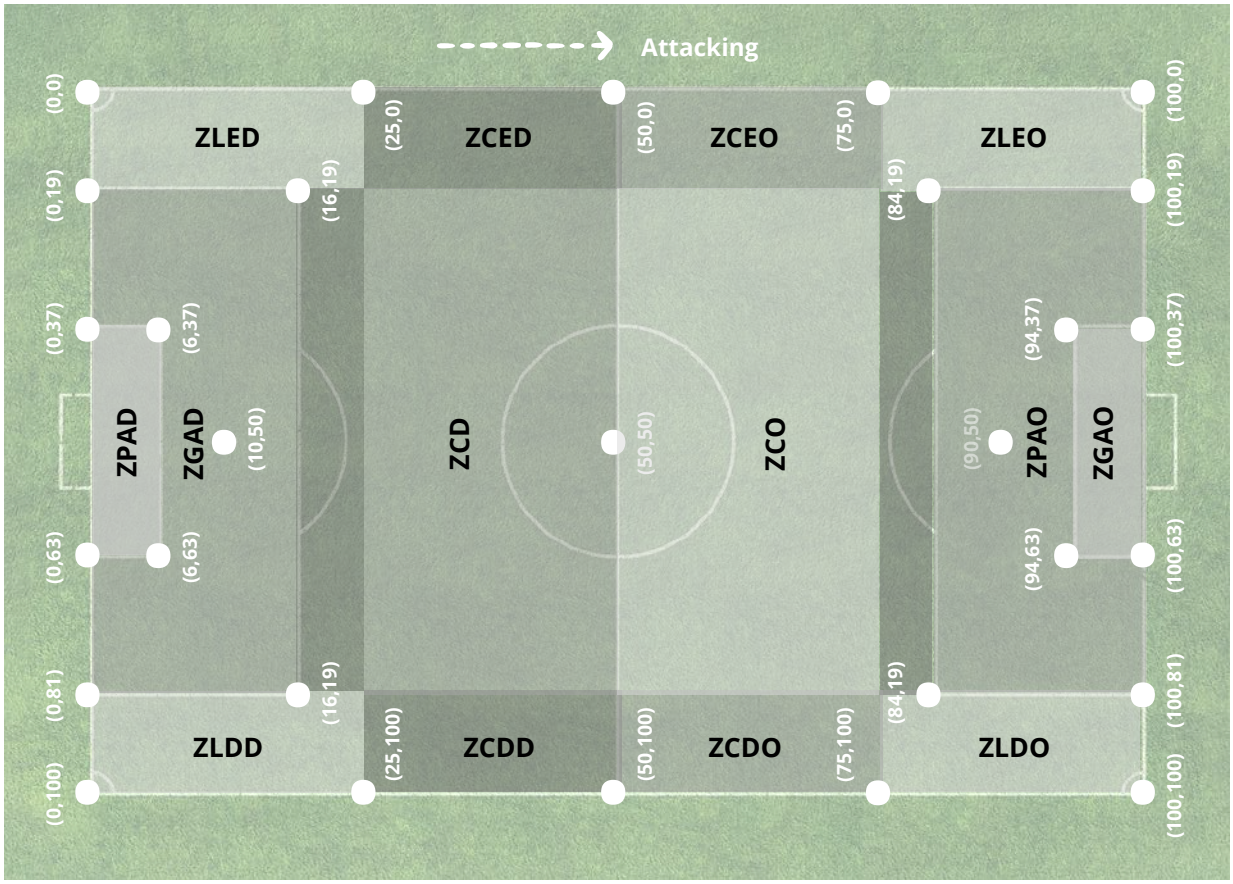


Figura 4.2: Segmentação do campo em 16 zonas, incluindo Pequena Área, Grande Área, Zona Central e Zonas Laterais, abrangendo as metades ofensiva e defensiva.

Outros tratamentos

Nos eventos de cobrança de falta com finalização, chute e pênalti, incluiu-se o resultado da tentativa como uma variável adicional, indicando se o lance resultou em gol (isto é, variável marcada como *goal*) ou em finalização malsucedida (isto é, variável marcada como *missed*). Essa definição baseia-se nas especificações apresentadas no glossário da *Wyscout*, segundo o qual apenas esses eventos podem resultar em gols. Além disso, gols contra não foram considerados.

Por fim, os nomes dos jogadores e de suas equipes foram concatenados em uma única palavra para facilitar a identificação, uma vez que existem atletas com nomes idênticos ou que atuaram por mais de uma equipe durante o campeonato.

4.3 Visão geral da metodologia

Dada a natureza estruturada dos dados de eventos de partidas de futebol e visando possibilitar a aplicação de modelos de linguagem, este estudo adota uma abordagem de geração de linguagem natural (do inglês *data-to-text generation*, DTG), na qual dados tabulares são transformados em descrições textuais (Gatt and Krahmer, 2018; Wiseman et al., 2017). Nesse enquadramento, cada registro de evento deixa de ser tratado apenas como uma observação em formato "linha–coluna" e passa a ser expresso como uma unidade textual com ordem e contexto. Assim, os eventos das partidas são convertidos em sentenças padronizadas, o que permite explorar a dimensão sequencial e relacional do jogo por meio de técnicas de Processamento de Linguagem Natural, preservando informações como o tipo de ação, os agentes envolvidos e a localização do lance em uma forma adequada ao tratamento linguístico.

Para realizar essa conversão, foi utilizada uma estratégia baseada em regras com o auxílio de *templates*, devido à sua interpretabilidade, reprodutibilidade e ao alinhamento direto com a estrutura dos dados de eventos esportivos. Em particular, os *templates* permitem controlar explicitamente quais variáveis compõem cada sentença e em que ordem elas aparecem, reduzindo ambiguidades e garantindo consistência na forma como diferentes tipos de evento são descritos (Reiter and Dale, 2000). Embora esse método possa levar a certa repetição lexical, ele garante alta precisão e consistência semântica, aspectos fundamentais para assegurar que as descrições geradas correspondam fielmente aos dados originais. Além disso, essa estruturação favorece futuras extensões com métodos mais flexíveis ou baseados em redes neurais (Moryossef et al., 2019; Liu et al., 2021).

A Figura 4.3 apresenta um exemplo do procedimento de geração de texto adotado neste estudo. A partir de um registro tabular de eventos **(A)**, aplica-se um *template* com campos pré-definidos **(B)** para gerar sentenças padronizadas **(C)**, que posteriormente são normalizadas para compor o *corpus* utilizado nas análises de NLP **(D)**.

A partir dos textos produzidos, são obtidas representações distribuídas por meio de dois paradigmas: *embeddings* estáticos (GloVe) e *embeddings* contextualizados (BERT). Em seguida, essas representações são exploradas em análises distintas, definidas de acordo com as particularidades de cada modelo: enquanto os *embeddings* estáticos fornecem uma representação única por termo, adequada para comparações globais e sínteses por vocabulário, os *embeddings* contextualizados permitem avaliar variações dependentes do contexto e padrões em nível de ocorrência. Dessa forma, as etapas analíticas são conduzidas separadamente para cada paradigma, respeitando suas propriedades e objetivos, e os resultados são apresentados em seções específicas.

4.4 Conversão de dados tabulares em texto

Com o objetivo de possibilitar a aplicação de técnicas de Processamento de Linguagem Natural, os dados estruturados das partidas de futebol foram convertidos para um formato textual. Essa transformação permite representar cada evento como uma sentença padronizada, incorporando informações essenciais do registro original (ação, entidade envolvida e contexto espacial), de modo que as descrições possam ser exploradas por modelos de *embeddings* e outras tarefas de NLP.

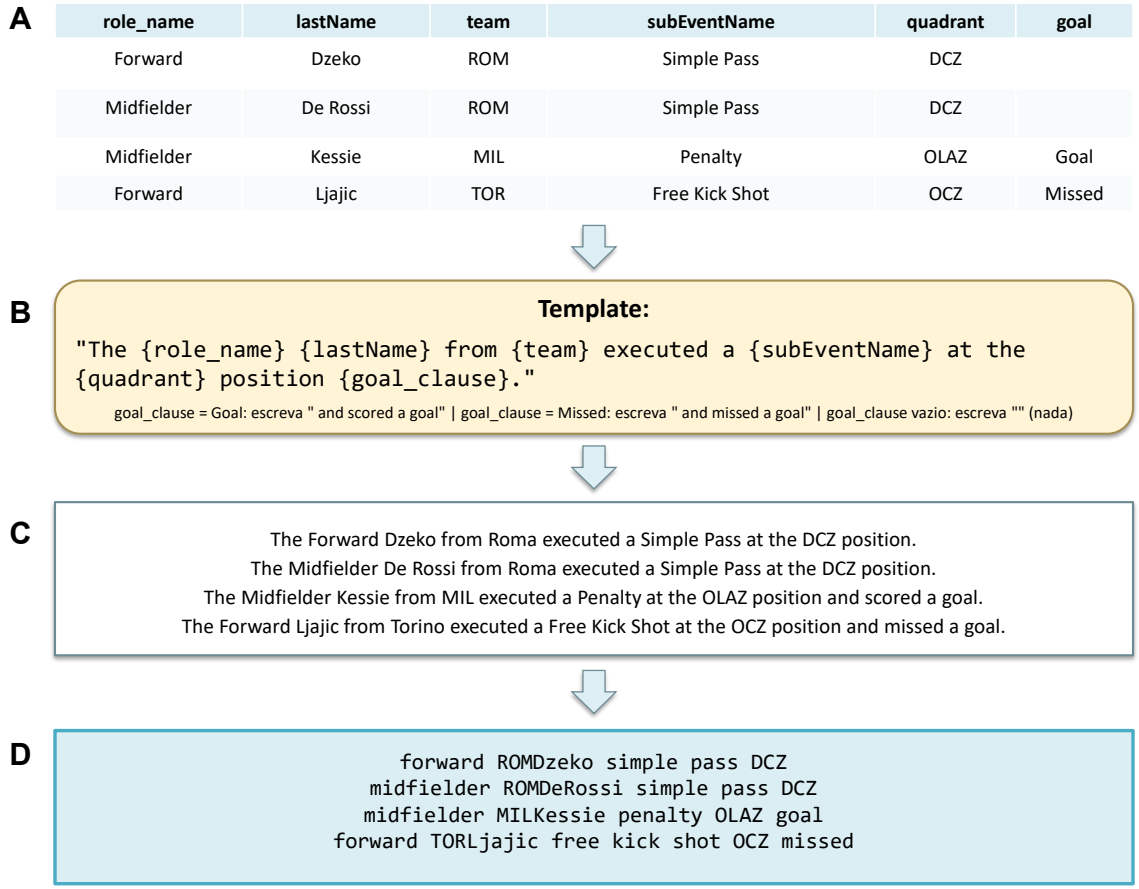


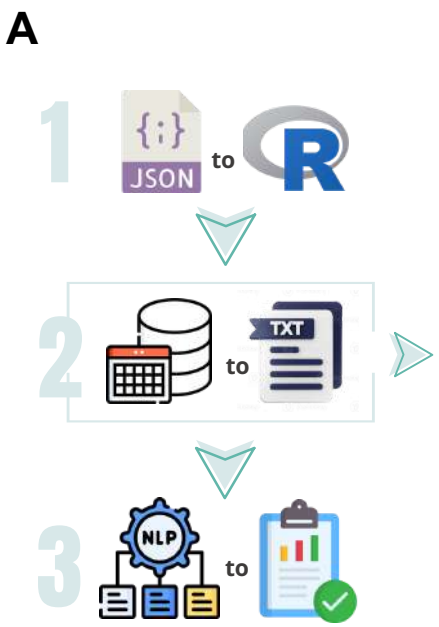
Figura 4.3: Ilustração da abordagem DTG: etapas da geração de texto a partir de eventos: **(A)** tabela, **(B)** template, **(C)** sentenças e **(D)** normalização.

A Figura 4.4 apresenta dois fluxos distintos. Em **(A)**, são descritas as etapas gerais: inicialmente, o conjunto de dados original em formato JSON é importado para o ambiente R (passo 1); em seguida, os dados tabulares são transformados em uma representação textual (passo 2); e, por fim, constrói-se uma estrutura adequada à modelagem baseada em NLP (passo 3). Em **(B)**, detalham-se as operações do passo 2: o processo de transformação foi concebido para identificar e priorizar as variáveis mais relevantes do conjunto de dados (etapa 1), permitindo sua interpretação como uma narrativa padronizada de cada evento da partida (etapa 2). Essa narrativa segue uma estrutura predefinida que incorpora posição, jogador/time, tipo de evento, posicionamento e resultado. A etapa final produz o texto processado, pronto para análises subsequentes de NLP, após a aplicação de procedimentos de pré-processamento e normalização (etapa 3).

4.5 Extração e análise dos embeddings

4.5.1 Modelo GloVe

As palavras do *corpus* foram *tokenizadas* considerando cada partida como um documento distinto, identificado pelo identificador da partida. A partir dessa *tokenização*, construiu-se um vocabulário



B

Stage 1						
matchID	**role_name**	**lastName**	**team**	**subEventName**	**quadrant**	**goal**
2575959	Forward	Dzeko	ROM	Simple Pass	DCZ	
2575959	Midfielder	De Rossi	ROM	Simple Pass	DCZ	
2575961	Midfielder	Kessie	MIL	Penalty	OLAZ	Goal
2575960	Forward	Ljajic	TOR	Free Kick Shot	OCZ	Missed

Stage 2
The **Forward Dzeko** from **Roma** executed a **Simple Pass** at the **DCZ** position.
The **Midfielder De Rossi** from **Roma** executed a **Simple Pass** at the **DCZ** position.
The **Midfielder Kessie** from **Milan** executed a **Penalty** at the **OLAZ** position and **scored a goal**
The **Forward Ljajic** from **Torino** executed a **Free Kick Shot** at the **OCZ** position and **missed a goal**

Stage 3
forward ROMDzeko simple pass DCZ
midfielder ROMDeRossi simple pass DCZ
midfielder MILKessie Penalty OLAZ goal
forward TORLjajic Free Kick Shot OCZ missed

Figura 4.4: Fluxo de *data-to-text generation* (DTG) adotado no estudo. Em **(A)**, descrevem-se as etapas gerais desde a importação do JSON até a construção do *corpus* para NLP. Em *(B)*, exemplifica-se a conversão de registros tabulares (*Stage* 1) em sentenças padronizadas por *templates* (*Stage* 2) e a etapa de pré-processamento e normalização do texto (*Stage* 3) para uso em modelos de *embeddings*.

inicial de 629 termos distintos, posteriormente reduzido para 611 após a exclusão de palavras com frequência inferior a cinco ocorrências, de forma a eliminar termos pouco informativos e reduzir ruído.

Com base no vocabulário resultante, foi gerada a matriz de coocorrência (do inglês *Term Co-occurrence Matrix*, TCM), considerando janelas de contexto de tamanho 5. O parâmetro de ponderação $x_{\max}$ foi definido como a razão entre o número de termos do vocabulário e 100, de modo a balancear a influência de coocorrências raras e frequentes.

O modelo foi ajustado com dimensão latente de 75, ao longo de 1.000 iterações, adotando tolerância de convergência de 10^{-5}. O resultado consistiu na obtenção de vetores densos de 75 dimensões para cada termo do vocabulário.

4.5.2 Modelo BERT

Para obter as representações distribuídas contextualizadas, optou-se pela utilização do modelo pré-treinado `BERT-base-multilingual-cased`, disponibilizado pela biblioteca *Transformers* da `Hugging Face` (Wolf et al., 2020). Esse modelo baseia-se na arquitetura do codificador *Transformer* (Vaswani et al., 2023) e foi treinado em *corpora* multilíngues.

Pré-treinamento adicional e ajuste do vocabulário

Foi realizado um pré-treinamento adicional do modelo BERT com o objetivo de adaptá-lo ao domínio do futebol. Para esse processo, utilizou-se o *tokenizador* do `BERT-base-multilingual-cased`, ao qual foram incorporados novos *tokens* identificados no conjunto de dados, ampliando assim o vocabulário do *tokenizador*. O treinamento do *tokenizador* foi realizado sobre o *corpus* extraído, utilizando a biblioteca `Hugging Face Tokenizers`, com vocabulário definido em 50.000 *tokens* e frequência mínima de ocorrência igual a 1. Essa etapa buscou garantir que o modelo fosse capaz de processar o vocabulário especializado, incluindo nomes de jogadores, posições e eventos relacionados às partidas.

O pré-treinamento foi conduzido utilizando a tarefa de *Masked Language Modeling* (MLM), em que 15% dos *tokens* no texto de entrada foram mascarados e o modelo foi treinado para prever os *tokens* ocultos (Devlin et al., 2019). O treinamento foi realizado por cinco épocas, com tamanho de lote igual a 8, utilizando o otimizador padrão da biblioteca `Hugging Face` e hiperparâmetros ajustados. O modelo foi implementado no *framework* `PyTorch` e o treinamento foi executado em GPU, visando reduzir o custo computacional.

Tokenização WordPiece e agregação de subpalavras

Inicialmente, os textos foram *tokenizados* por meio do *tokenizador* associado ao modelo, o qual emprega o algoritmo **WordPiece** para segmentar palavras em subunidades lexicais (Wu et al., 2016). Como essa segmentação pode resultar em múltiplos *tokens* para uma única palavra, implementou-se um procedimento de agregação de *embeddings*. Nessa etapa, as subpalavras correspondentes à mesma palavra foram agrupadas, e suas representações vetoriais foram combinadas por meio da média, produzindo um único *embedding* representativo por palavra (Devlin et al., 2019).

Segmentação de sequências com sobreposição

Devido à limitação arquitetural do BERT, que permite processar sequências de no máximo 512 *tokens* (Devlin et al., 2019), adotou-se uma estratégia de segmentação com janelas sobrepostas. Considere a sequência de *tokens* de um texto

$$T = (t_1, t_2, \ldots, t_N).$$

Dividiu-se T em segmentos S_k de comprimento máximo $L_{\max}$ com sobreposição de O *tokens* entre segmentos consecutivos. O k-ésimo segmento é definido como:

$$S_k = \left(t_{(k-1)(L_{\max}-O)+1}, \ldots, t_{(k-1)(L_{\max}-O)+L_{\max}}\right),$$

onde $k = 1, 2, \ldots, K$, em que K representa o número total de segmentos necessários para cobrir todos os N *tokens*, calculado por:

$$K = \left\lceil \frac{N - O}{L_{\max} - O} \right\rceil.$$

No presente estudo, utilizou-se $L_{\max} = 500$ e $O = 10$. Essa parametrização assegura que cada segmento compartilhe os últimos O *tokens* com o anterior, preservando continuidade contextual. Essa técnica, análoga ao método de janela deslizante, busca minimizar perda semântica nas fronteiras entre segmentos (Sun et al., 2019; Beltagy et al., 2020).

Extração dos embeddings contextualizados

Cada segmento foi processado individualmente pelo modelo, e as representações da última camada oculta foram extraídas como *embeddings* contextualizados. Esses vetores foram armazenados juntamente com seus respectivos *tokens*, compondo uma base de representações distribuídas utilizada nas análises subsequentes. Como as projeções são dependentes do contexto, cada ocorrência de um termo no *corpus* está associada a um vetor potencialmente distinto, refletindo o conteúdo contextual em que o termo aparece (Rogers et al., 2021).

4.6 Análise das redes ponderadas

Para cada equipe, foi construída uma rede não direcionada e ponderada, representada por um grafo $G = (V, E)$, no qual o conjunto de vértices V corresponde aos jogadores e o conjunto de arestas E representa as relações entre pares de jogadores. A cada aresta (i, j) foi associado um peso contínuo w_{ij}, interpretado como a intensidade da relação entre os jogadores i e j. As análises foram conduzidas separadamente para cada equipe.

Os pesos das arestas foram definidos a partir da similaridade de cosseno entre representações vetoriais associadas aos jogadores, conforme apresentado na Equação 2.6. Como a similaridade de cosseno assume valores no intervalo $[-1, 1]$, foi aplicado um deslocamento aditivo (*shift*) de uma unidade, de modo que os pesos das arestas fossem estritamente não negativos:

$$w_{ij} = \cos(\vec{v}, \vec{w}) + 1.$$

Essa transformação preserva a ordenação relativa das similaridades, ao mesmo tempo em que garante valores positivos para os pesos, condição necessária para a aplicação de métricas de centralidade ponderadas e para a interpretação dos pesos como intensidades de conexão.

Redução de densidade por backbone adaptativo baseado em percentis

Para reduzir a complexidade estrutural e facilitar a interpretação visual, foi empregada uma estratégia de *backbone* adaptativo baseada em percentis por nó.

Para cada vértice $i \in V$, seja $\{w_{ij}\}_{j \in \mathcal{N}(i)}$ o conjunto de pesos das arestas incidentes em i. Define-se um limiar específico τ_i como o percentil p dessa distribuição, com $p = 80$ no presente estudo. Uma aresta (i, j) foi mantida no *backbone* se

$$w_{ij} \geq \tau_i \quad \text{ou} \quad w_{ij} \geq \tau_j,$$

caracterizando um critério de união. Essa abordagem evita a imposição de um limiar global fixo e reduz o viés associado a vértices de alto grau, preservando conexões localmente relevantes mesmo para jogadores com menor conectividade. Após a aplicação do *backbone*, vértices isolados foram removidos do grafo.

O uso de *backbones* adaptativos em redes ponderadas densas é amplamente discutido na literatura de redes complexas como alternativa metodologicamente mais flexível a limiares globais (Serrano et al., 2009; Tumminello et al., 2011).

Layout e codificação visual

A disposição espacial dos vértices foi obtida por meio de um algoritmo de *layout* baseado em forças, no qual os pesos das arestas influenciam a proximidade entre os nós no plano (Fruchterman and Reingold, 1991). Essa estratégia produz representações nas quais vértices fortemente conectados tendem a ocupar regiões próximas.

Os vértices foram coloridos de acordo com a posição funcional do jogador em campo: atacantes foram destacados na cor roxa, meio-campistas na cor azul, defensores na cor verde e, por fim, goleiros na cor amarela. O tamanho dos vértices foi definido proporcionalmente ao grau ponderado (*strength*), permitindo identificar visualmente jogadores com maior intensidade total de conexões. As arestas foram representadas com largura e transparência proporcionais ao peso w_{ij}, de modo que conexões mais intensas se destacassem visualmente, enquanto relações de menor magnitude permanecessem atenuadas.

Medidas de centralidade

A importância estrutural dos jogadores foi avaliada por meio de duas métricas clássicas de centralidade em redes ponderadas: grau ponderado e centralidade de intermediação.

O **grau ponderado (strength)** de um vértice i foi definido como

$$s_i = \sum_j w_{ij},$$

representando a intensidade total das conexões do jogador na rede. Essa medida generaliza o conceito de grau em grafos ponderados e é amplamente utilizada para quantificar a relevância de vértices em redes com pesos contínuos (Barrat et al., 2004; Newman, 2010).

Já a **centralidade de intermediação (betweenness)** foi calculada a partir da transformação dos pesos em distâncias, definida como

$$d_{ij} = \frac{1}{w_{ij}},$$

de modo que conexões mais intensas correspondam a caminhos mais curtos no grafo. Essa centralidade quantifica a fração de caminhos mínimos entre todos os pares de vértices que passam por um determinado nó, identificando jogadores que atuam como intermediários estruturais entre diferentes subconjuntos da rede (Freeman, 1977; Newman, 2010).

Para cada equipe, foram identificados os jogadores com maior valor de *strength* e maior valor de *betweenness*, interpretados como vértices dominantes segundo os critérios de centralidade. Esses vértices foram destacados visualmente nos grafos por meio de símbolos específicos: ★ para *strength* e ♢ para *betweenness*.

Similaridade estrutural e agrupamento hierárquico

Para comparar as estruturas globais das redes de jogadores entre as equipes, foi utilizado o *NetLSD*, que resume propriedades espectrais do grafo em um vetor descritivo baseado no espectro do Laplaciano

normalizado (Tsitsulin et al., 2018). A distância entre duas equipes foi então definida como a distância euclidiana entre seus respectivos vetores NetLSD, resultando em uma matriz de distâncias simétrica entre todas as equipes analisadas.

A partir dessa matriz de distâncias, realizou-se um agrupamento hierárquico utilizando o método de ligação média (*average linkage*), no qual a distância entre dois grupos é definida como a média das distâncias entre todos os pares de elementos pertencentes aos respectivos grupos. Essa hierarquia foi representada por meio de um dendrograma, e a matriz de distâncias NetLSD foi reordenada de acordo com a ordem das folhas do dendrograma, permitindo a visualização conjunta das similaridades estruturais e da organização hierárquica das equipes.

A definição do número de clusters foi inicialmente orientada pelo índice de *silhouette* médio, calculado para diferentes valores de k, considerando a matriz de distâncias NetLSD como entrada. O índice de *silhouette* avalia simultaneamente a coesão intra-cluster e a separação inter-cluster, assumindo valores mais elevados quando as observações estão bem alocadas aos seus respectivos grupos.

4.7 UMAP e HDBSCAN para inspeção dos embeddings contextualizados

Como parte da etapa exploratória, avaliou-se a estrutura dos *embeddings* contextualizados em nível de partida por meio de uma combinação de redução de dimensionalidade e agrupamento não supervisionado. Para cada jogo, partiu-se da matriz de representações por termo, após a agregação das ocorrências, na qual cada palavra é associada a um vetor em alta dimensão. Em seguida, aplicou-se o UMAP para projetar esses vetores em duas dimensões, de forma a preservar as relações de proximidade no espaço vetorial e permitir a análise visual.

Paralelamente, utilizou-se o algoritmo HDBSCAN para identificar agrupamentos densos de termos semanticamente próximos, sem fixar previamente o número de grupos. O HDBSCAN foi escolhido por permitir que observações com baixa densidade sejam classificadas como ruído, reduzindo a formação de *clusters* artificiais (Campello et al., 2013).

O UMAP foi parametrizado para considerar 15 vizinhos na definição da vizinhança local e uma distância mínima de 0,1 na projeção; o HDBSCAN foi aplicado com um tamanho mínimo de cluster igual a 7, permitindo também a classificação de observações como ruído.

Capítulo 5

Resultados

Foram realizadas análises considerando representações distribuídas obtidas tanto pelo modelo **BERT** quanto pelo **GloVe**. Os resultados derivados de cada abordagem serão apresentados a seguir, com ênfase nas características próprias de cada metodologia e nos aspectos que surgem a partir de sua aplicação.

5.1 Modelo GloVe

A Figura 5.1 apresenta a projeção bidimensional das representações vetoriais dos clubes, obtidas por meio dos *embeddings* oriundos do modelo GloVe, utilizando o UMAP para redução de dimensionalidade.

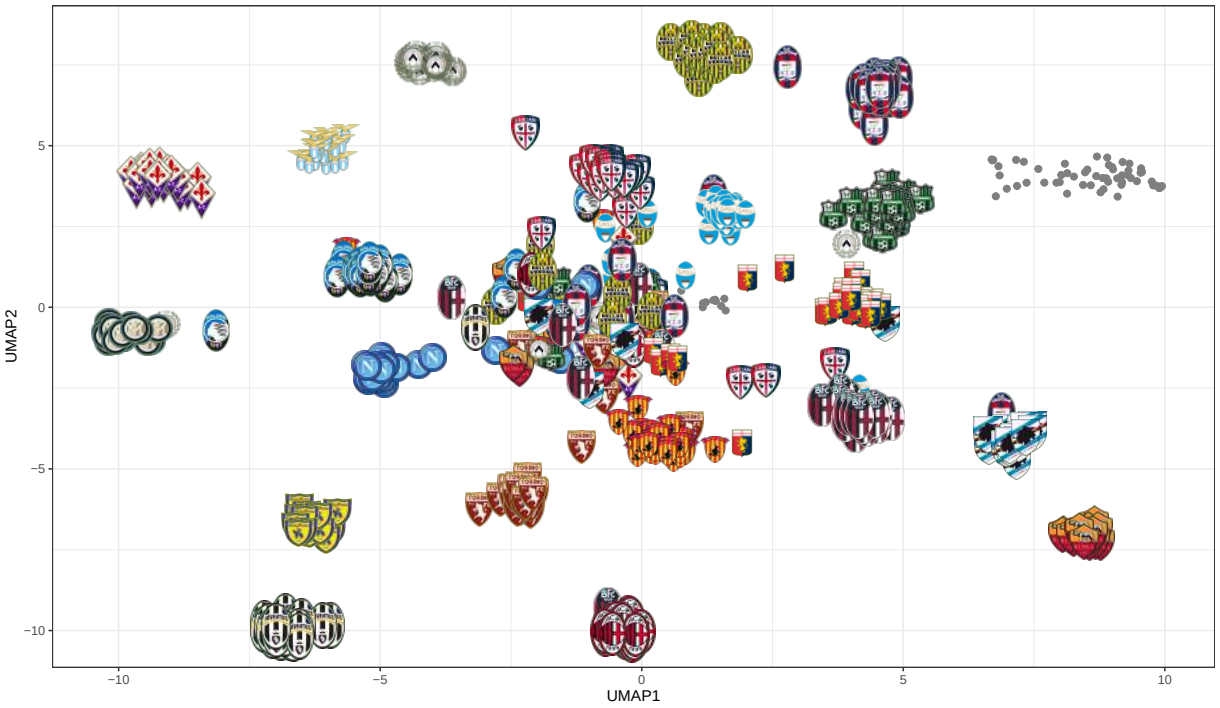


Figura 5.1: Visualização UMAP dos embeddings GloVe dos jogadores da Série A Italiana (2017–18).

Cada ponto no gráfico corresponde a um *token*, sendo os jogadores representados pelos escudos dos respectivos clubes, enquanto outras palavras, como posições em campo e tipos de eventos, aparecem apenas na cor cinza. No geral, a visualização evidencia padrões distintos de agrupamento entre as equipes, sugerindo que, em média, jogadores do mesmo time apresentam representações vetoriais semelhantes.

Alguns clubes aparecem de forma mais isolada, posicionando-se nas extremidades do espaço bidimensional. Esse afastamento sugere que seus *embeddings* apresentam menor similaridade em relação aos demais, o que pode refletir diferenças no estilo de jogo ou particularidades dos eventos associados a essas equipes. Entre os clubes mais destacados, encontram-se os principais da liga italiana, como Juventus, Milan e Internazionale, cujos elencos tendem a se diferenciar pela maior consistência e relativamente maior homogeneidade interna.

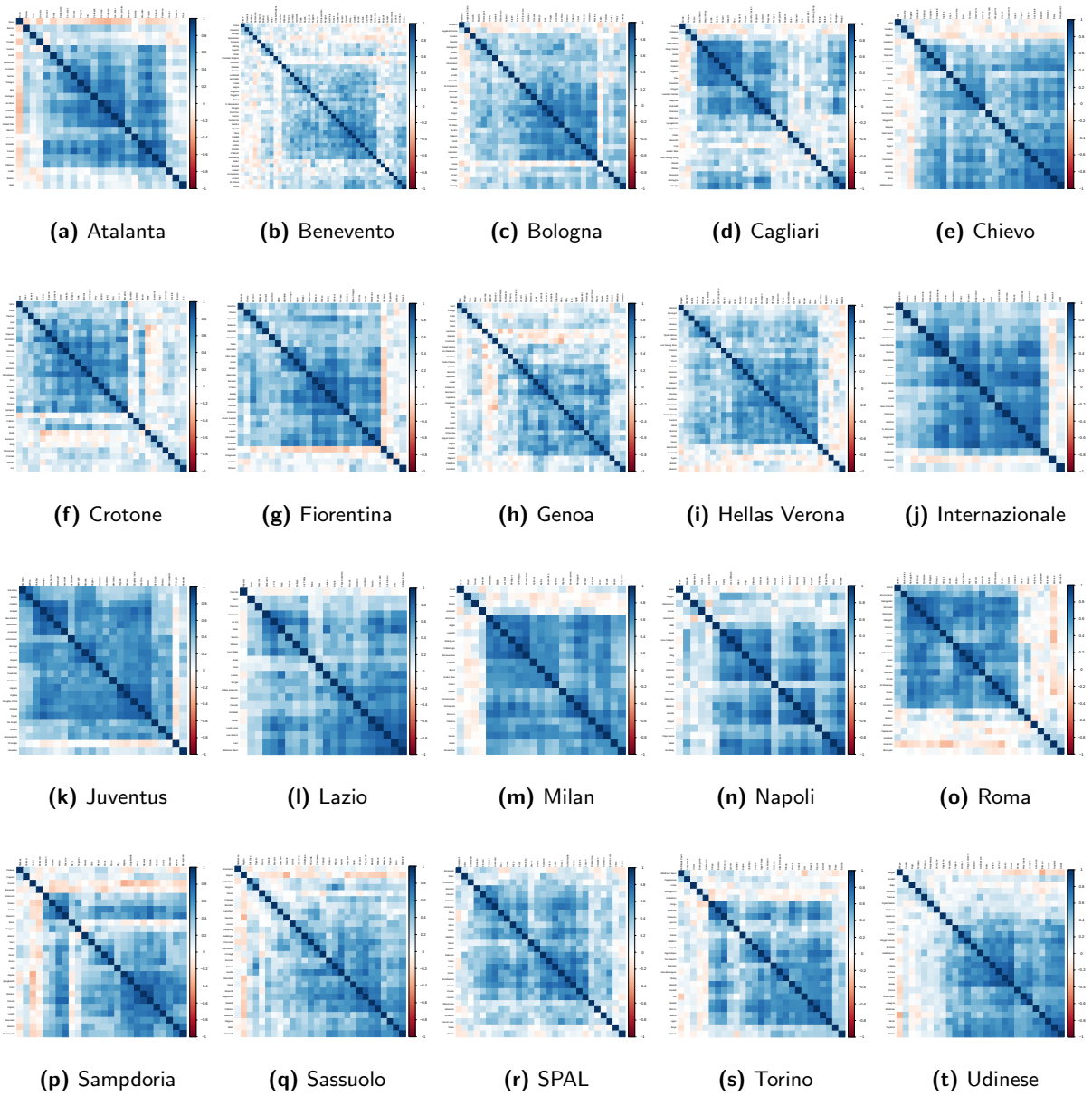
No centro da Figura 5.1 observa-se uma região mais densa, com sobreposição de escudos. Esse padrão sugere maior proximidade semântica entre os *embeddings*, indicando perfis de jogadores bastante semelhantes e formando um núcleo comum. Agrupamentos localizados próximos a essa região central, mas com algum grau de separação, configuram agrupamentos intermediários: equipes que mantêm certas características próprias, embora ainda compartilhem elementos estruturais com outros clubes.

Matrizes de Similaridade de Cosseno

Adicionalmente, como forma de medir a similaridade entre os jogadores de uma mesma equipe, a Figura 5.2 apresenta as matrizes de similaridade de cosseno entre atletas de cada clube da Série A italiana da temporada analisada, construídas a partir dos *embeddings* do modelo GloVe. Cada célula da matriz corresponde à similaridade do cosseno entre um par de jogadores. Esse valor varia teoricamente entre -1 e 1 e expressa o ângulo entre os vetores associados aos jogadores no espaço de *embeddings*. Valores próximos de 1 indicam maior similaridade, enquanto valores próximos de 0 sugerem pouca ou nenhuma relação, e valores negativos refletem vetores orientados em direções opostas. Na escala de cores utilizada, tons de azul-escuro representam maior similaridade, enquanto tonalidades avermelhadas indicam menor similaridade. A diagonal principal assume valor igual a 1, por se tratar da comparação de cada jogador consigo mesmo.

Nota-se que alguns clubes apresentam padrões mais definidos. O Napoli (2º) Figura 5.2n, e a Lazio (5º) Figura 5.2l, por exemplo, exibem blocos bem delimitados de alta similaridade, o que sugere a presença de grupos de jogadores com funções semelhantes ou fortemente relacionadas em contexto. Em contraste, equipes como o Benevento (20º) Figura 5.2b e o Genoa (12º) Figura 5.2h, apresentam matrizes mais heterogêneas, com menor formação de blocos, indicando maior diversidade semântica entre os jogadores e papéis possivelmente mais distribuídos dentro do elenco.

Em geral, equipes de melhor desempenho tendem a apresentar matrizes de similaridade mais homogêneas, sugerindo um padrão de interações mais consistente, enquanto equipes de pior desempenho exibem matrizes mais fragmentadas, compatíveis com maior variabilidade de funções e relações. Além disso, jogadores com baixa similaridade em relação ao restante do time frequentemente têm menor tempo de jogo na temporada, acumulando menos eventos e, portanto, representações menos estáveis.



(a) Atalanta **(b)** Benevento **(c)** Bologna **(d)** Cagliari **(e)** Chievo

(f) Crotone **(g)** Fiorentina **(h)** Genoa **(i)** Hellas Verona **(j)** Internazionale

(k) Juventus **(l)** Lazio **(m)** Milan **(n)** Napoli **(o)** Roma

(p) Sampdoria **(q)** Sassuolo **(r)** SPAL **(s)** Torino **(t)** Udinese

Figura 5.2: Matrizes de similaridade de cosseno entre jogadores da Série A Italiana (2017–18) com *embeddings* do Modelo GloVe.

A Figura 5.3 apresenta a matriz de similaridade de cosseno entre os jogadores da Roma, a mesma apresentada em Figura 5.2o mas com destaque, construída a partir das representações vetoriais geradas pelo modelo de NLP.

Observa-se a formação de blocos com altos valores de similaridade entre jogadores de linha, indicando padrões semânticos semelhantes nas descrições das ações em jogo, o que sugere uma dinâmica coletiva consistente na construção ofensiva e defensiva da equipe.

Jogadores de perfil defensivo e de meio-campo como Pellegrini (meia), Cristante (volante), Mancini (zagueiro), Smalling (zagueiro), Spinazzola (lateral) e Karsdorp (lateral) apresentam elevada proximidade vetorial, possivelmente associada à frequência de interações diretas e à participação em

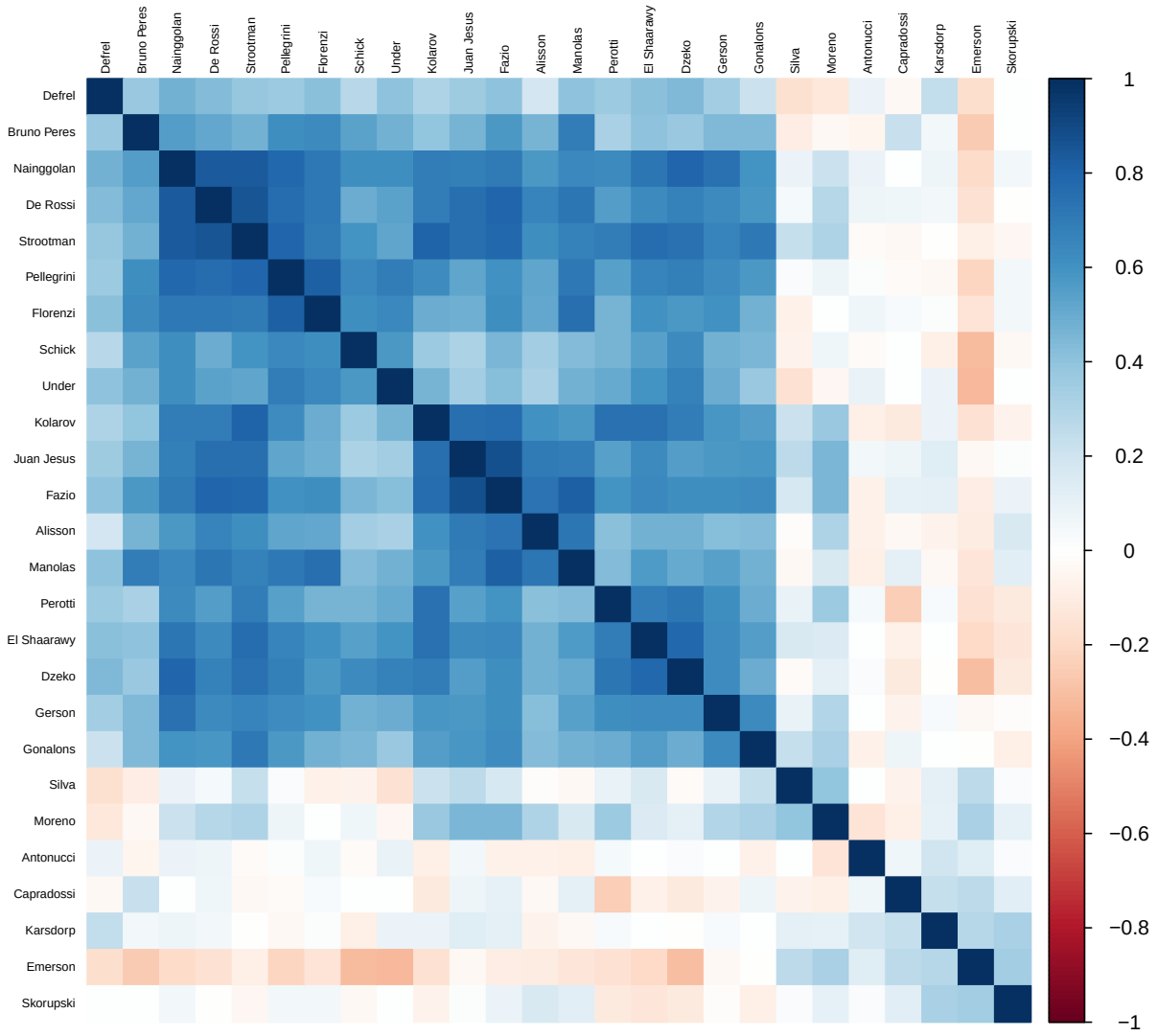


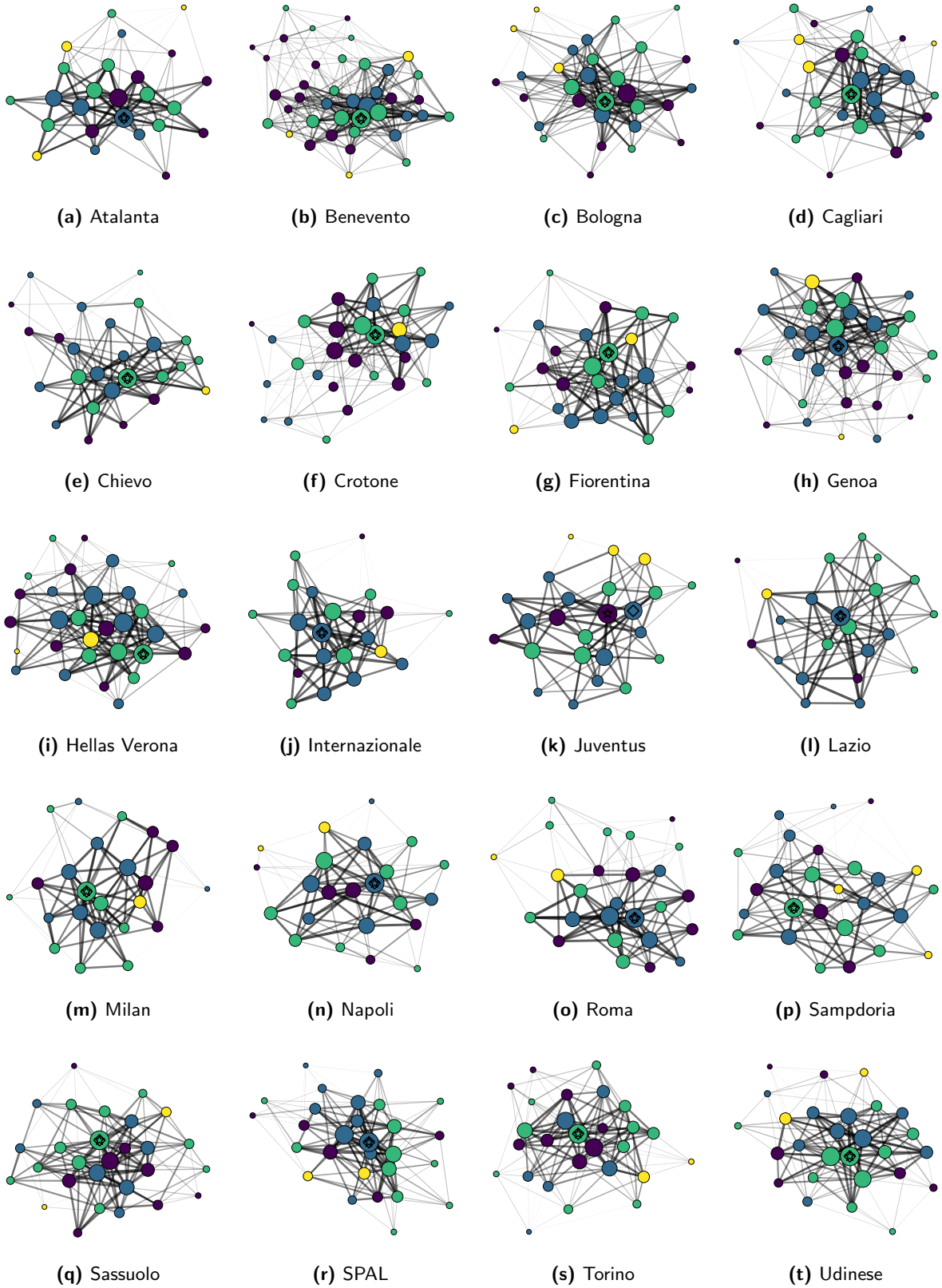
Figura 5.3: Padrões estruturais na matriz de similaridade de cosseno da Roma (Figura 5.2o)

sequências recorrentes de passes, coberturas e transições. De forma semelhante, atacantes como Džeko e El Shaarawy mostram maior similaridade com meias ofensivos, como Zaniolo e Mkhitaryan, refletindo padrões de associação entre criação e finalização de jogadas. Em contraste, os goleiros, como Alisson e Skorupski, apresentam menores níveis de similaridade em relação aos jogadores de linha, o que era esperado devido à natureza distinta de suas ações em campo.

Redes Ponderadas

A Figura 5.4 apresenta as redes ponderadas de similaridade entre jogadores para as equipes analisadas após a aplicação do *backbone* adaptativo baseado no percentil $p = 80$. De modo geral, observa-se que as redes exibem estruturas bem conectadas, com núcleos centrais densos e redução significativa de conexões periféricas. A distribuição do grau ponderado é assimétrica, com poucos jogadores concentrando grande parte da intensidade das conexões, enquanto a maioria apresenta valores

moderados.



(a) Atalanta **(b)** Benevento **(c)** Bologna **(d)** Cagliari

(e) Chievo **(f)** Crotone **(g)** Fiorentina **(h)** Genoa

(i) Hellas Verona **(j)** Internazionale **(k)** Juventus **(l)** Lazio

(m) Milan **(n)** Napoli **(o)** Roma **(p)** Sampdoria

(q) Sassuolo **(r)** SPAL **(s)** Torino **(t)** Udinese

Figura 5.4: Redes de jogadores por clube na Serie A italiana, categorizadas por posição em campo do Modelo GloVe.

Para cada equipe, destacaram-se os jogadores com maior *strength* e maior *betweenness*, interpretados como vértices dominantes por grau ponderado e centralidade de intermediação. Esses vértices são indicados por ★ (*strength*) e ♢ (*betweenness*). Os nós foram coloridos pela posição funcional: atacantes em roxo, meio-campistas em azul, defensores em verde e goleiros em amarelo.

Ao comparar as redes entre equipes, observa-se que times de melhor desempenho apresentam estruturas mais homogêneas, com núcleos centrais compostos por vários jogadores com valores próximos de centralidade. Esse padrão é observado, por exemplo, em equipes como Juventus (1º) e Internazionale (4º) apresentam redes nas quais a centralidade se distribui de maneira mais equilibrada entre múltiplos jogadores centrais, indicando maior coesão e compartilhamento de protagonismo estrutural. Em contraste, equipes de desempenho inferior na tabela, como Benevento (20º), Crotone (18º) e Chievo (13º), tendem a exibir maior concentração estrutural em um único vértice dominante, logo a rede depende mais fortemente de um jogador específico, evidenciando maior vulnerabilidade estrutural à remoção desse vértice.

A Tabela 5.1 resume, para cada equipe, os jogadores centrais segundo o grau ponderado (*strength*) e a centralidade de intermediação (*betweenness*), incluindo os valores das métricas e as posições dos jogadores. Em praticamente todas as equipes analisadas, o jogador identificado como principal segundo o critério de grau coincide com o jogador de maior centralidade. Esse resultado indica que, nessas redes, a intensidade total das conexões está fortemente associada à posição estrutural do jogador nos caminhos mínimos do grafo.

Tabela 5.1: Jogadores centrais por equipe segundo grau ponderado (strength) e centralidade de intermediação (betweenness).

Clube	**Strength**			**Betweenness**		
	Jogador	**Valor**	**Posição**	**Jogador**	**Valor**	**Posição**
Atalanta	Freuler	23,38	Meio-campista	Freuler	0,14	Meio-campista
Benevento	Letizia	39,83	Defensor	Letizia	0,17	Defensor
Bologna	Masina	32,06	Defensor	Masina	0,13	Defensor
Cagliari	Ceppitelli	29,94	Defensor	Ceppitelli	0,15	Defensor
Chievo	Cacciatore	28,16	Defensor	Cacciatore	0,20	Defensor
Crotone	Sampirisi	23,50	Defensor	Sampirisi	0,10	Defensor
Fiorentina	Astori	26,88	Defensor	Astori	0,14	Defensor
Genoa	Laxalt	28,21	Meio-campista	Laxalt	0,10	Meio-campista
Hellas Verona	Caracciolo	23,70	Defensor	Caracciolo	0,12	Defensor
Internazionale	Perisic	22,09	Meio-campista	Perisic	0,19	Meio-campista
Juventus	Mandzukic	19,47	Atacante	Matuidi	0,16	Meio-campista
Lazio	Milinkovic Savic	30,59	Meio-campista	Milinkovic Savic	0,24	Meio-campista
Milan	Rodriguez	21,69	Defensor	Rodriguez	0,16	Defensor
Napoli	Jorginho	21,86	Meio-campista	Jorginho	0,13	Meio-campista
Roma	Florenzi	21,64	Meio-campista	Florenzi	0,17	Meio-campista
Sampdoria	Silvestre	22,34	Defensor	Silvestre	0,12	Defensor
Sassuolo	Acerbi	30,08	Defensor	Acerbi	0,19	Defensor
SPAL	Lazzari	29,97	Meio-campista	Lazzari	0,14	Meio-campista
Torino	De Silvestri	28,02	Defensor	De Silvestri	0,14	Defensor
Udinese	Samir	27,85	Defensor	Samir	0,17	Defensor

Observa-se ainda uma predominância de meio-campistas e defensores entre os principais nós, o que é consistente com o fato de essas posições estarem, em geral, mais envolvidas em múltiplas fases de organização, como construção e circulação e, portanto, exibirem maior diversidade de relações com diferentes setores do time; isso aumenta tanto a soma das similaridades (*strength*) quanto a probabilidade de ocupar caminhos mínimos na rede (*betweenness*).

Os jogadores identificados como centrais são, em geral, compatíveis com a organização estrutural de suas respectivas equipes. Em times de maior porte, como Napoli (2°) e Lazio (5°), a centralidade de meio-campistas como Jorginho e Milinković-Savić reflete modelos de jogo nos quais a circulação e a progressão ofensiva ocupam papel central, favorecendo atletas responsáveis pela conexão entre setores e pela organização das interações.

Em contraste, em equipes de menor porte, como Benevento (20º) e Cagliari (16º), a identificação de defensores como principais nós é consistente com estruturas mais orientadas à contenção e à organização defensiva, nas quais a linha defensiva assume maior protagonismo na sustentação e na distribuição das interações, resultando em maior conectividade estrutural desses jogadores.

A Figura 5.5 apresenta os histogramas das métricas por jogador para a Juventus (1°) e o Benevento (20°), considerando o grau ponderado e a centralidade de intermediação.

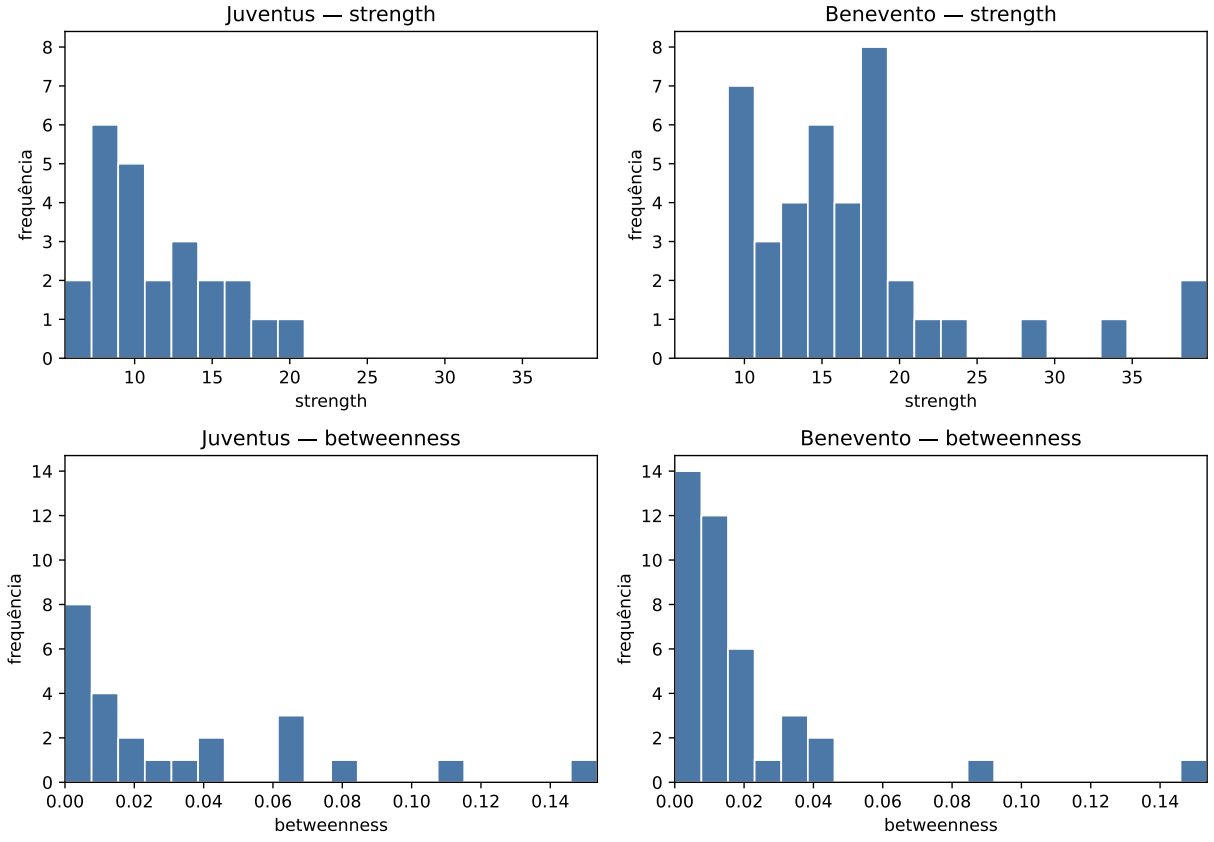


Figura 5.5: Distribuição das métricas de centralidade na Juventus e no Benevento: campeã *vs.* último colocado

Em relação ao grau ponderado, observa-se que a Juventus concentra a maior parte dos jogadores em valores relativamente baixos a moderados, sem a presença de caudas muito longas, sugerindo uma distribuição mais homogênea da intensidade total de conexões dentro da equipe. Em contraste, o

Benevento apresenta uma distribuição mais dispersa e com valores extremos mais elevados, indicando maior heterogeneidade e possível concentração estrutural de conexões em poucos jogadores.

Para a centralidade de intermediação, ambas as equipes exibem distribuição fortemente assimétricas à direita, com grande parte dos jogadores próxima de zero, o que é esperado em redes nas quais poucos nós concentram caminhos mínimos entre pares de vértices. Ainda assim, o Benevento tende a apresentar maior concentração em valores muito próximos de zero e alguns jogadores com valores relativamente mais altos, enquanto a Juventus mostra uma dispersão ligeiramente maior na faixa intermediária.

Em conjunto, os resultados sugerem que a equipe com pior desempenho apresenta uma estrutura mais dependente de jogadores específicos para conectar diferentes partes da rede, ao passo que a equipe campeã apresenta uma organização estrutural relativamente menos concentrada.

NetLSD

A Figura 5.6 apresenta a matriz de similaridade entre as redes de jogadores dos clubes da Série A, calculada a partir do descritor NetLSD, o qual permite comparar a estrutura dos grafos independentemente do tamanho das equipes. Valores menores indicam maior similaridade entre as redes, enquanto valores maiores refletem diferenças estruturais mais acentuadas.

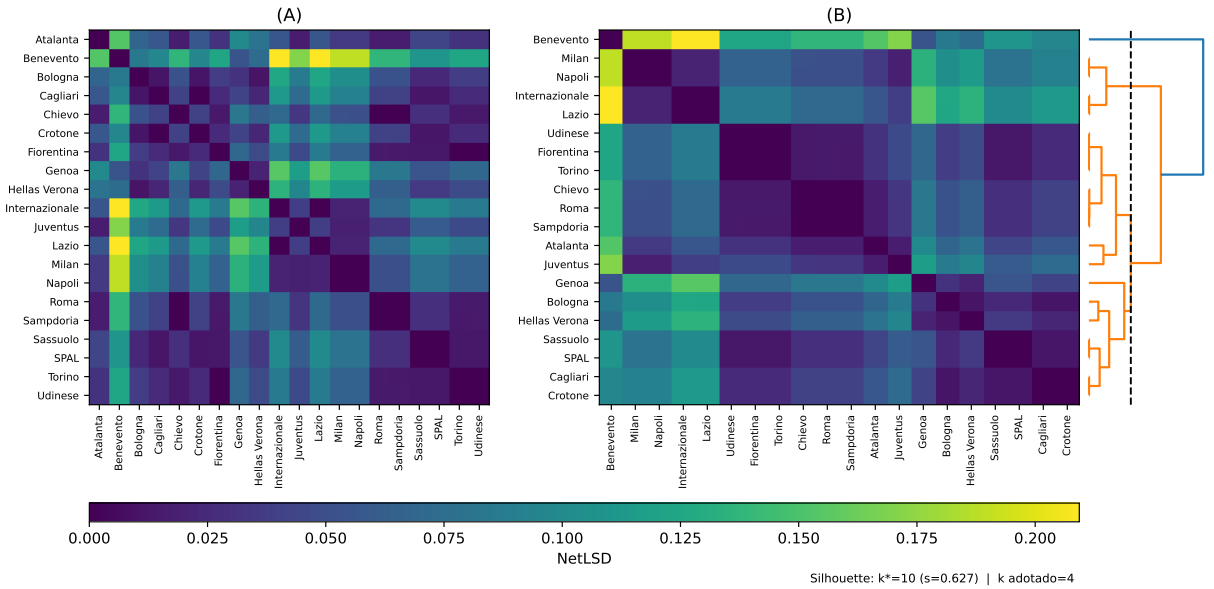


Figura 5.6: (A) Matriz de distâncias e **(B)** agrupamento hierárquico das redes de equipes com base no NetLSD nos *embeddings* do Modelo GloVe.

Na matriz de distâncias NetLSD em **(A)**, observa-se uma heterogeneidade considerável nas similaridades estruturais, indicada pela variação das cores ao longo da matriz. Algumas equipes, como Napoli, Lazio e Internazionale, exibem distâncias relativamente menores entre si, sugerindo padrões estruturais próximos, enquanto outras, como Crotone, Cagliari e SPAL, apresentam maiores distâncias em relação à maioria dos demais clubes. Essas características se alinham ao desempenho dessas equipes na competição, conforme pode ser observado na Tabela 4.1.

Em **(B)**, a matriz é reordenada de acordo com o agrupamento hierárquico, permitindo a identificação de blocos estruturais coerentes, definidos a partir do corte adotado no dendrograma. Embora o valor máximo do índice de *silhouette* tenha sido observado para um número relativamente elevado de *clusters*, a curva apresentou comportamento não monotônico e ganhos marginais para valores crescentes de k, especialmente em função do número limitado de equipes analisadas. Dessa forma, visando maior parcimônia e interpretabilidade, adotou-se uma partição com $k = 4$ *clusters*. O nível de corte correspondente foi identificado no dendrograma e indicado graficamente por uma linha tracejada, a partir da qual foram definidos os agrupamentos finais dos clubes.

A reordenação da matriz segundo o agrupamento hierárquico, com corte em $k = 4$ *clusters*, revela uma separação estrutural consistente entre as equipes. O primeiro *cluster* reúne Internazionale ($4°$), Napoli ($2°$), Milan ($6°$) e Lazio ($5°$), agrupando equipes que finalizaram na parte superior da tabela e que apresentam alta similaridade mútua, com redes mais homogêneas. Um segundo agrupamento engloba Sampdoria ($10°$), Roma ($3°$), Juventus ($1°$), Torino ($9°$), Atalanta ($7°$), Fiorentina ($8°$), Chievo ($13°$) e Udinese ($14°$), indicando padrões intermediários de organização estrutural e maior diversidade de posições ao longo da classificação. O terceiro *cluster* concentra Genoa ($12°$), Crotone ($18°$), Cagliari ($16°$), Bologna ($15°$), Sassuolo ($11°$), SPAL ($17°$) e Hellas Verona ($19°$), reunindo predominantemente equipes da metade inferior da tabela, cujas estruturas se mostram mais próximas entre si e mais afastadas das do primeiro grupo. Por fim, Benevento ($20°$) constitui isoladamente um quarto agrupamento, refletindo uma configuração estrutural singular.

5.2 Modelo BERT

Nas análises com BERT, foram utilizados os *embeddings* contextualizados gerados a partir do *corpus* de partidas, armazenados separadamente por partida. Como esse tipo de representação atribui um vetor a cada ocorrência de uma palavra, dependendo do contexto em que ela aparece, tornou-se necessário sumarizar essas ocorrências para viabilizar comparações e análises em nível de termo. Assim, para cada uma das 380 partidas, as representações foram agregadas por palavra pela média, obtendo-se um único vetor representativo por termo.

Análise de Partidas

A Figura 5.7 apresenta a projeção bidimensional (UMAP) dos termos de quatro partidas do campeonato, juntamente com os agrupamentos destacados pelo HDBSCAN. Observa-se um padrão consistente de organização dos *tokens* em regiões que refletem papéis distintos na estrutura textual gerada pelos *templates*.

É possível identificar: (i) um agrupamento predominantemente composto por jogadores de ambas as equipes, (ii) um agrupamento associado à localização no campo (zonas) e (iii) um conjunto mais disperso relacionado ao vocabulário de ações e eventos. Esse comportamento é compatível com a construção do *corpus*, na qual jogadores e zonas aparecem de forma altamente padronizada e recorrente nas sentenças, enquanto *tokens* de ação ocorrem em combinações contextuais mais variadas, resultando em menor densidade e, frequentemente, em classificação como ruído pelo algoritmo de agrupamento.

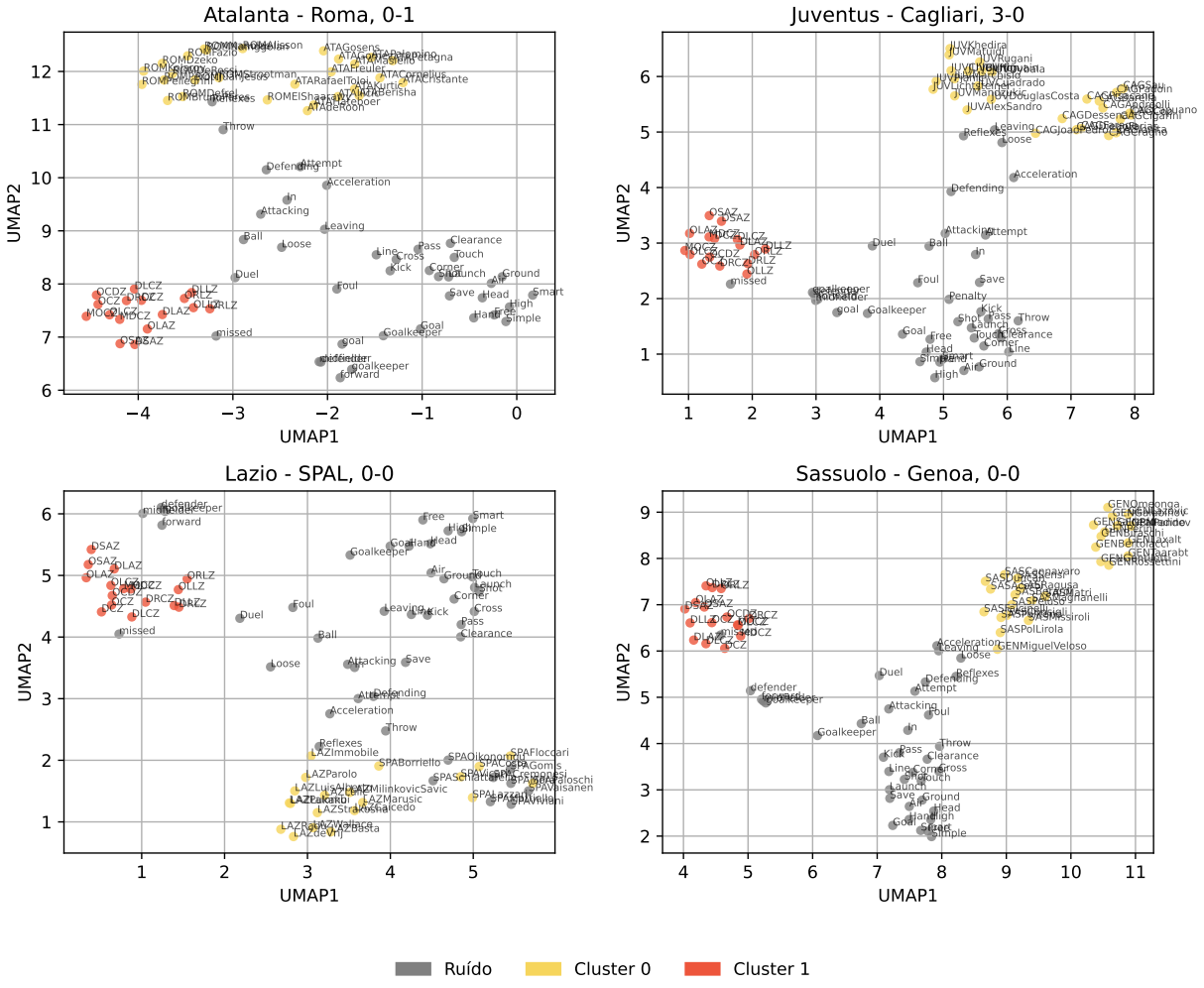


Figura 5.7: UMAP + HDBSCAN: estrutura dos *tokens* em partidas com diferentes desfechos.

Nas quatro partidas analisadas, há perfis bem diferentes de confronto: Juventus vs. Cagliari (3–0) representa um jogo com resultado mais confortável, típico de maior controle ao longo do tempo. Atalanta vs. Roma (0–1) sugere um duelo decidido por margem mínima, em que um evento pode ser determinante. Já Lazio vs. SPAL (0–0) e Sassuolo vs. Genoa (0–0) indicam jogos mais travados no placar, compatíveis com cenários de equilíbrio ou baixa efetividade ofensiva.

Ainda sobre os tipos de resultado, a goleada (**Juventus vs. Cagliari, 3–0**) apresenta um agrupamento mais compacto e bem separado das zonas, sugerindo maior homogeneidade nas representações associadas aos termos centrais do jogo, ou seja, jogadores e ações recorrentes, enquanto o conjunto de ações permanece intermediário, conectando regiões do mapa. Já na vitória simples (**Atalanta vs. Roma, 0–1**), observa-se maior sobreposição e continuidade entre o bloco de jogadores e parte do vocabulário de eventos, o que é compatível com uma partida mais equilibrada em termos de diversidade de ocorrências e contextos associados às ações descritas.

Nos empates (**Lazio vs. SPAL, 0–0; Sassuolo vs. Genoa, 0–0**), a estrutura geral se mantém, mas o gráfico tende a mostrar o núcleo de ações ocupando uma região relativamente maior e mais dispersa, com os jogadores distribuídos de forma menos polarizada. Esse padrão é compatível com jogos sem desfecho de gol, nos quais o texto pode ser dominado por sequências de eventos e transições

(passes, disputas diretas pela bola no chão ou pelo alto, recuperações, entre outros), sem que um subconjunto pequeno de termos se torne extremamente característico do resultado.

Em conjunto, esses resultados sugerem que os *embeddings* contextualizados capturam não apenas proximidade lexical, mas também informação funcional do jogo codificada no *template*: quem participa da ação (jogador), onde ela ocorre (zona do campo) e o que é realizado (tipo de evento). Na prática, isso se traduz em aproximar eventos e combinações que desempenham papéis semelhantes na dinâmica da partida. Por exemplo, ações associadas à construção e circulação de bola, à transição e disputa pela posse, ou à progressão e criação de chances.

A Figura 5.8 apresenta mapas de calor da similaridade de cosseno calculada entre pares de jogadores das mesmas quatro partidas apresentadas anteriormente. Em cada figura, linhas e colunas correspondem aos jogadores, identificados por prefixos do time, e a matriz é visualmente particionada em quatro quadrantes: os blocos na diagonal (superior esquerdo e inferior direito) representam as similaridades intra-equipe, enquanto os blocos fora da diagonal capturam similaridades entre equipes.

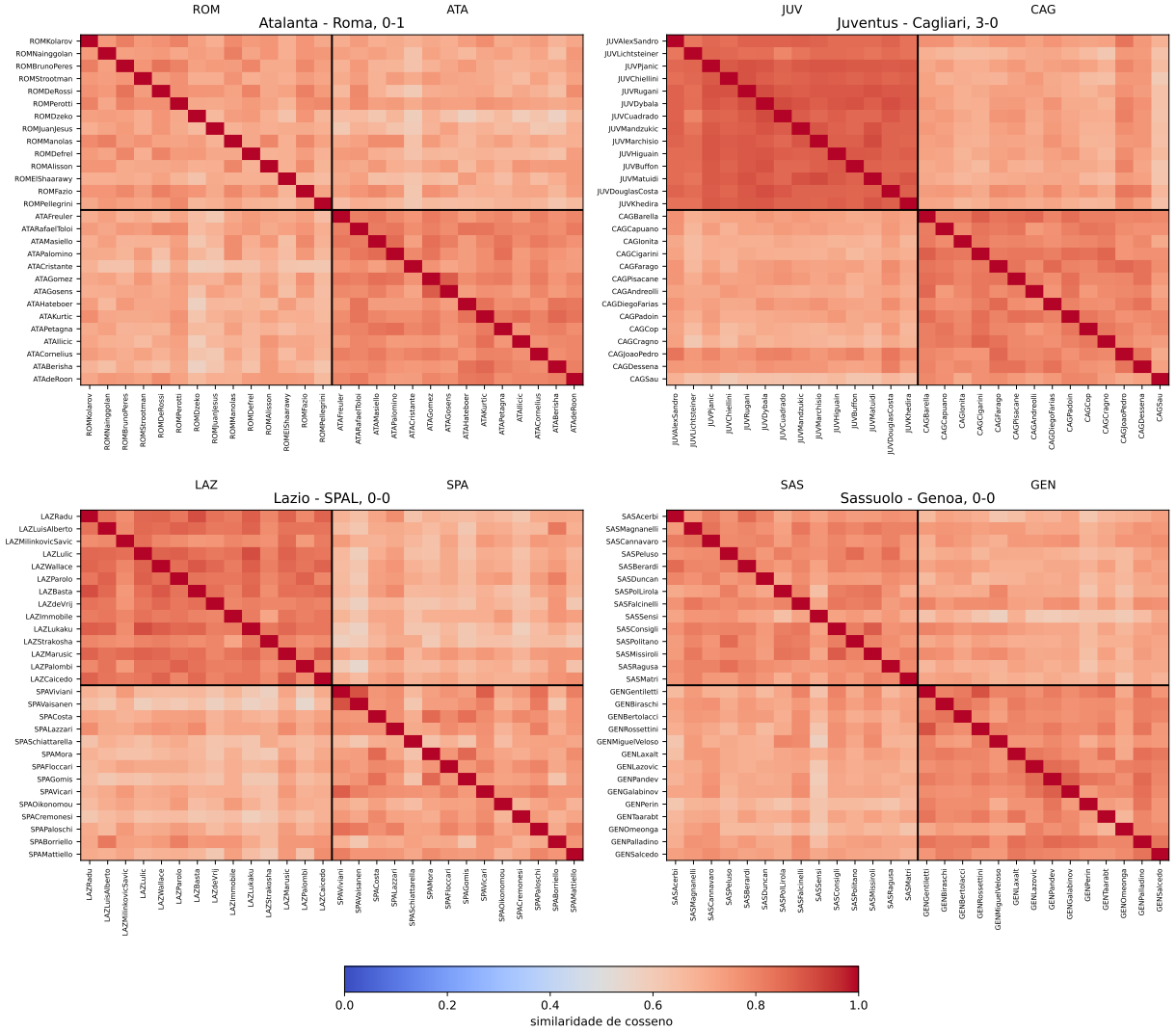


Figura 5.8: Similaridade de cosseno entre *embeddings* médios (BERT) de jogadores em quatro partidas.

De modo geral, os blocos intra-equipe tendem a concentrar valores mais altos do que os blocos

entre equipes, sugerindo que os jogadores de um mesmo time compartilham associações mais próximas dentro da representação utilizada. Já as regiões entre equipes exibem maior heterogeneidade e, em muitos casos, menores intensidades, indicando menor proximidade entre jogadores adversários no espaço vetorial.

Além do contraste entre blocos intra e inter-equipes, observam-se concentrações internas que sugerem jogadores cujos *embeddings* médios se aproximam fortemente dos de seus companheiros, formando núcleos de maior coesão tática. Em Atalanta vs. Roma (0–1), essas concentrações aparecem de forma mais evidente no bloco da Atalanta (7°), com destaque para Alejandro Gómez (meia ofensivo) e Robin Gosens (lateral esquerdo), além da proximidade de Marten de Roon (volante), indicando alinhamento entre criação e transição. No bloco da Roma (3°), as intensidades são mais irregulares, com maior centralidade em Džeko (centroavante) e De Rossi (volante), refletindo papéis estratégicos na finalização e organização defensiva.

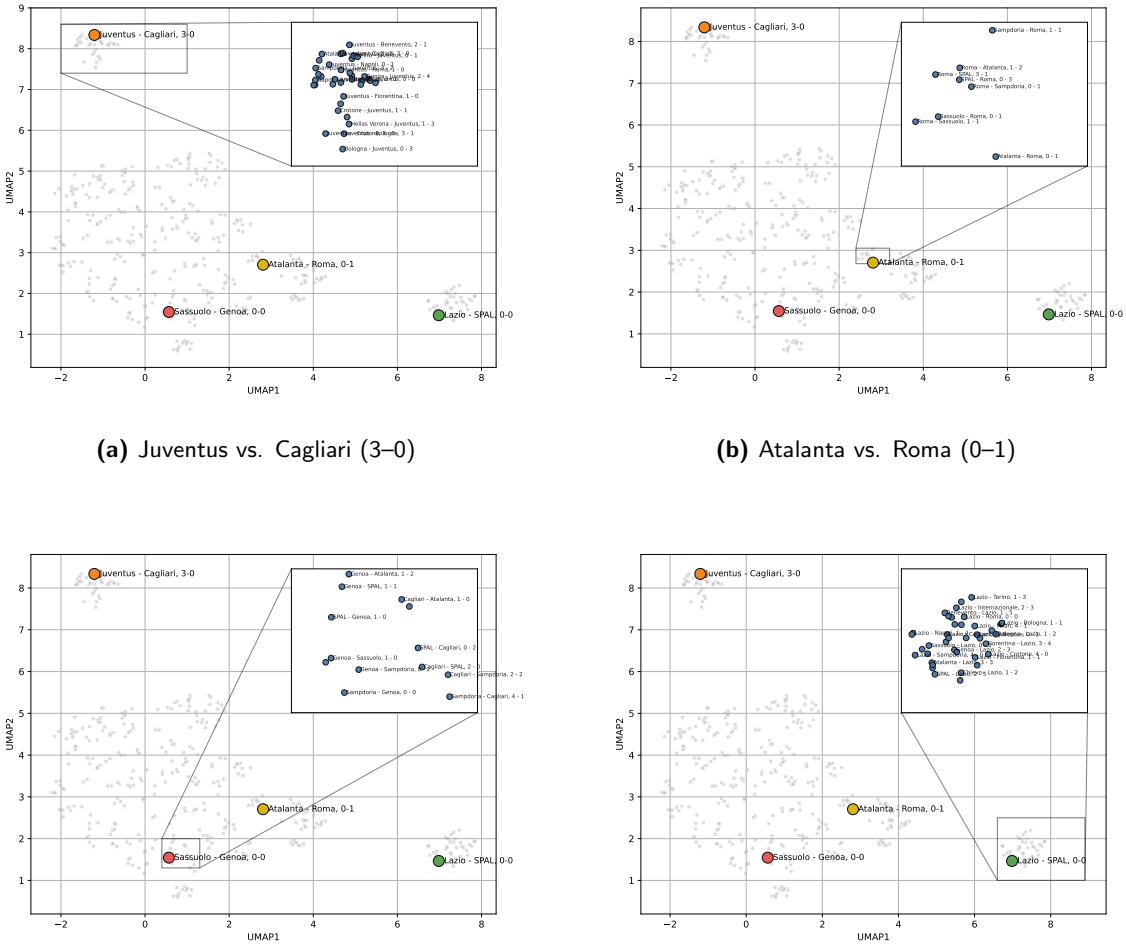
Em Juventus vs. Cagliari (3–0), o bloco intra-equipe da Juventus (1°) apresenta-se particularmente homogêneo e intenso, evidenciando forte integração entre Pjanić (meia), Dybala (meia atacante) e Higuaín (centroavante), coerente com o desempenho dominante da equipe campeã. Já o Cagliari (16°) exibe intensidades relativamente altas, porém mais heterogêneas, com concentrações pontuais em Barella (meia central) e João Pedro (atacante), sugerindo maior dependência de ações individuais dentro de uma estrutura coletiva menos coesa.

Nos empates sem gols, Lazio vs. SPAL e Sassuolo vs. Genoa, os resultados refletem tanto a organização posicional quanto o desempenho das equipes na temporada. Na Lazio (5°), o núcleo formado por Luis Alberto (meia ofensivo), Milinković-Savić (meia central) e Immobile (centroavante) concentra a dinâmica ofensiva. Já na SPAL (17°), as intensidades altas aparecem de forma mais pontual, destacando Lazzari (lateral) e Viviani (volante), compatíveis com um perfil mais defensivo e reativo.

De maneira semelhante, no confronto Sassuolo vs. Genoa, Berardi e Politano (pontas) e Sensi (meia) formam o principal eixo ofensivo do Sassuolo (11°), enquanto no Genoa (12°) a concentração recai sobre Veloso (meia) e Pandev (atacante).Esses padrões reforçam que, mesmo após a sumarização por média dos *embeddings* do BERT os *heatmaps* preservam não apenas a distinção entre equipes, mas também sinais associados às posições, funções táticas e ao nível de desempenho ao longo do campeonato.

A assinatura de cada partida foi obtida em duas etapas: primeiro, para cada jogador da partida, foi calculado um único vetor de *embedding* médio do *token* ao longo de suas ocorrências no jogo. Esses vetores formam, para cada partida, uma matriz em que cada linha é um jogador e cada coluna é uma dimensão do *embedding*. Segundo, para transformar essa matriz em uma representação única do jogo, foi calculada a mediana em cada dimensão, ou seja, coluna a coluna, resultando em um vetor resumo da partida.

A Figura 5.9 apresenta a projeção bidimensional das assinaturas das partidas obtidas com UMAP. O cálculo a partir dessas assinaturas permite visualizar a organização do conjunto no plano, de modo que partidas mais semelhantes tendem a ocupar posições mais próximas. No gráfico, os pontos em cinza representam todas as partidas e os quatro jogos que estão sendo analisados nos resultados são realçados com cores e rótulos.



(a) Juventus vs. Cagliari (3–0)

(b) Atalanta vs. Roma (0–1)

(c) Sassuolo vs. Genoa (0–0)

(d) Lazio vs. SPAL (0–0)

Figura 5.9: Partidas no espaço UMAP: distribuição global e vizinhanças de quatro jogos de referência.

Visualmente, as partidas distribuem-se de forma diferenciada no mapa, refletindo o desempenho das equipes ao longo do campeonato. O confronto Juventus vs. Cagliari (3–0) surge isolado no canto superior esquerdo, afastado do grande aglomerado central, sugerindo um comportamento mais característico associado à equipe campeã. Já Atalanta vs. Roma (0–1) posiciona-se em uma região intermediária do plano, próxima a jogos de equipes do pelotão superior. Os dois empates sem gols também se distinguem espacialmente: Sassuolo vs. Genoa (0–0) aparece mais abaixo, próximo à massa central, onde se concentram diversas partidas de equipes da metade inferior da tabela. Enquanto Lazio vs. SPAL (0–0) desloca-se para a direita, ocupando uma região menos densa do espaço.

No destaque verde, Lazio vs. SPAL (0–0) insere-se em uma área onde também se observam outras partidas da Lazio, indicando que, no espaço do UMAP, os jogos da equipe tendem a ocupar uma região relativamente consistente. Em contraste, no destaque laranja, os jogos da Juventus (1°) formam um agrupamento próprio e deslocado em relação ao aglomerado principal, evidenciando maior distinção em comparação à média das demais equipes do campeonato.

No destaque amarelo, Atalanta vs. Roma (0–1) aparece próximo de outras partidas da Roma,

sugerindo que o time, apresenta uma assinatura relativamente estável. Contudo, diferentemente da Juventus, essa regularidade não se traduz em um conglomerado completamente isolado, mas sim em uma região compartilhada com jogos de outras equipes do pelotão superior.

Por sua vez, no destaque vermelho, Sassuolo vs. Genoa (0–0) situa-se em um agrupamento mais heterogêneo, no qual aparecem partidas envolvendo equipes como SPAL (17°), Sampdoria (10°) e Cagliari (16°). Esse padrão sugere que confrontos entre equipes de desempenho intermediário ou da metade inferior da tabela tendem a apresentar assinaturas menos distintas, aproximando-se de um padrão mais comum dentro da competição.

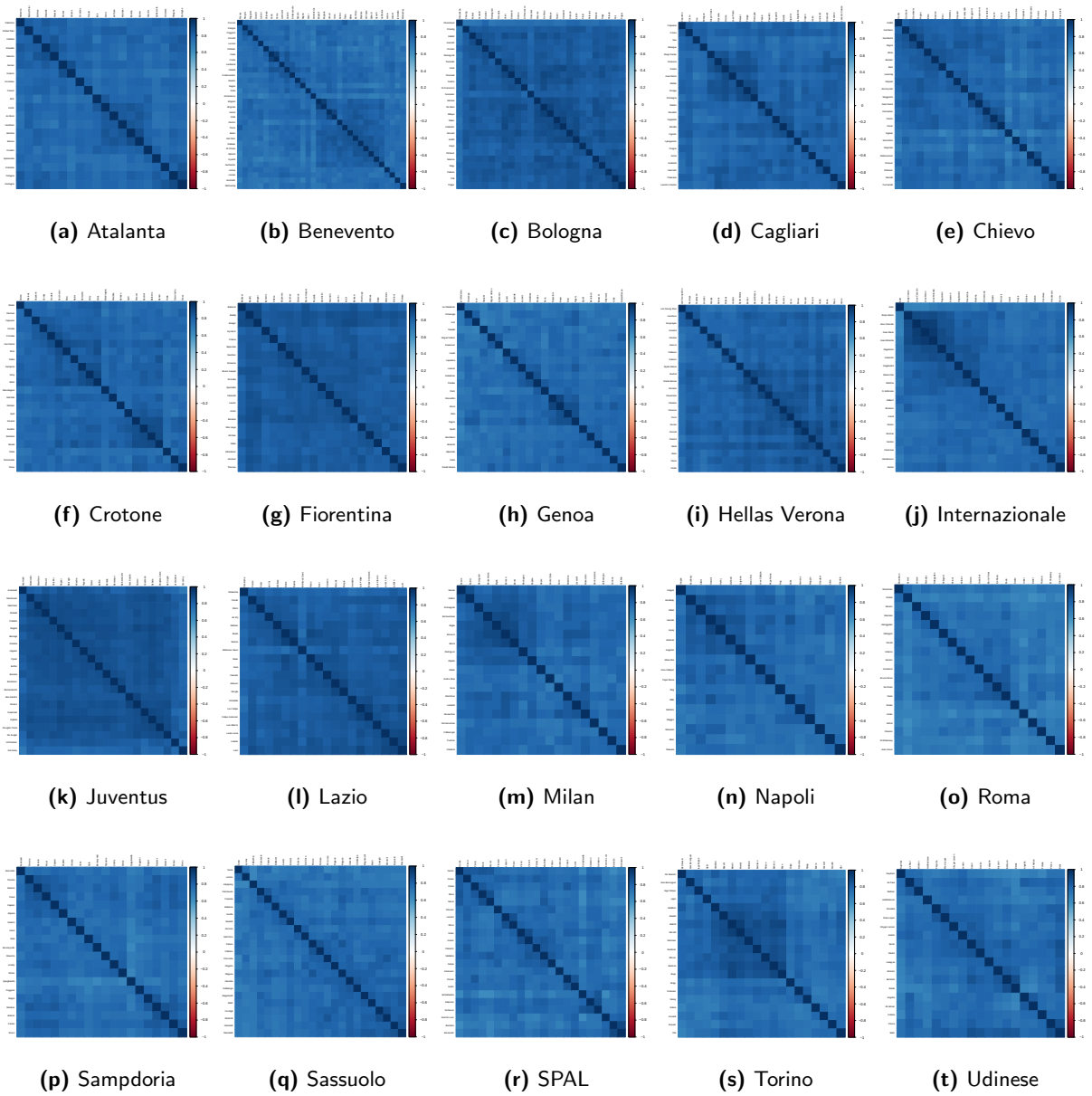
Matrizes de Similaridade de Cosseno

Para permitir a construção de redes de jogadores a partir de *embeddings* contextualizados, as representações do BERT foram agregadas em dois níveis. Inicialmente, em cada partida, múltiplas ocorrências de um mesmo *token* foram resumidas pela média, resultando em um vetor representativo por *token* no jogo. Em seguida, para obter uma representação global ao longo do campeonato, os vetores do mesmo jogador foram agregados entre partidas através da mediana, produzindo um único *embedding* por jogador na temporada. A partir desses vetores, para cada time foi estimada uma matriz de proximidade entre jogadores pela similaridade de cosseno.

A Figura 5.10 apresenta, para cada equipe, a matriz de similaridade entre jogadores obtida a partir das representações BERT agregadas ao longo da temporada. Tons mais escuros indicam maior similaridade entre os vetores dos jogadores, enquanto padrões mais homogêneos sugerem maior coesão global, e matrizes mais fragmentadas indicam maior heterogeneidade.

A Figura 5.10 evidencia que, após a etapa de agregação, média por partida e mediana ao longo do campeonato, as matrizes de similaridade por equipe tendem a apresentar valores relativamente altos de forma generalizada. Ao resumir os *embeddings* contextualizados de cada jogador em um vetor estável de temporada, parte da variabilidade situacional, dependente do contexto imediato de cada ocorrência, é suavizada, fazendo com que diferentes atletas do mesmo time acabem mais próximos no espaço vetorial. Em outras palavras, a agregação favorece uma representação mais global do padrão coletivo do time, o que comprime as distâncias e aumenta a similaridade média entre pares, resultando em matrizes com predominância de tons mais escuros e menor contraste extremo.

Na Roma, Figura 5.10o, apesar de a agregação ter deixado a matriz globalmente com similaridades relativamente altas, ainda é possível enxergar três padrões mais nítidos, ainda que leves: Alisson, Manolas e Nainggolan, Pellegrini e Perotti, e Bruno Peres, De Rossi e Fazio. Esses mini-blocos refletem papéis e contextos de participação que tendem a coocorrer nos eventos: no primeiro, a combinação de goleiro, defensor e um meio-campista de alto volume sugere um padrão ligado a saída de bola, cobertura e transições, em que ações defensivas e de progressão inicial aparecem de forma recorrente. No segundo, Pellegrini e Perotti concentram ações típicas de construção ofensiva e criação, participação em zonas e sequências parecidas, com envolvimento em passes e chances, o que aproxima suas representações e, no terceiro, Bruno Peres, De Rossi e Fazio formam um bloco ligado a circulação e sustentação da posse, ligações laterais e centralidade na organização, gerando *embeddings* mais próximos entre si do que em relação ao restante do elenco.



(a) Atalanta **(b)** Benevento **(c)** Bologna **(d)** Cagliari **(e)** Chievo

(f) Crotone **(g)** Fiorentina **(h)** Genoa **(i)** Hellas Verona **(j)** Internazionale

(k) Juventus **(l)** Lazio **(m)** Milan **(n)** Napoli **(o)** Roma

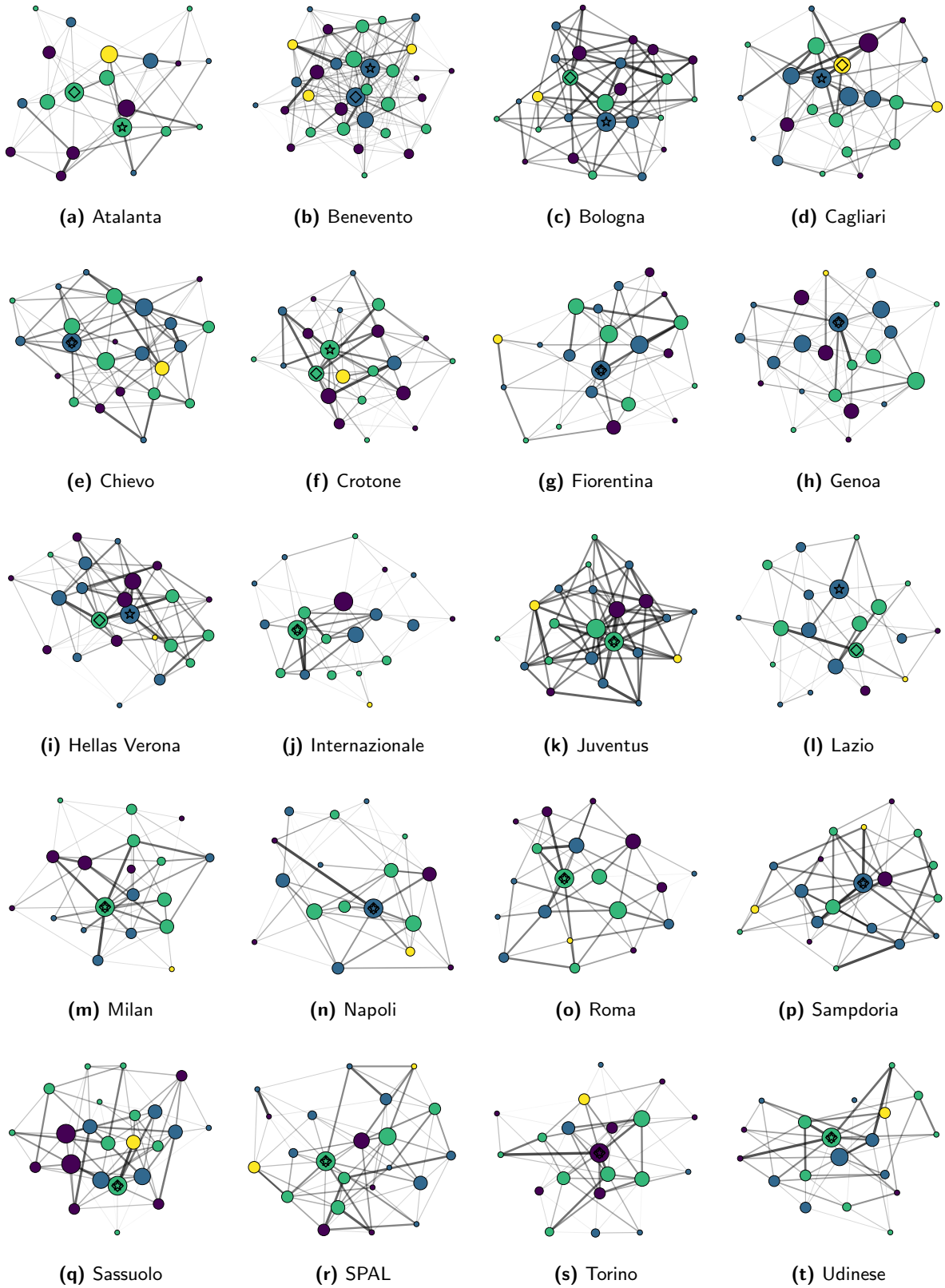
(p) Sampdoria **(q)** Sassuolo **(r)** SPAL **(s)** Torino **(t)** Udinese

Figura 5.10: Matrizes de similaridade de cosseno entre jogadores da Série A Italiana (2017–18) com *embeddings* do Modelo BERT.

Redes Ponderadas

As matrizes de similaridade de cosseno foram utilizadas para construir, para cada equipe, um grafo não direcionado e ponderado, em que os nós representam os jogadores e os pesos das arestas refletem a proximidade no espaço vetorial. A partir dessas redes, foram aplicadas as mesmas etapas descritas para o caso do GloVe, uma vez que o procedimento de construção e análise das redes permaneceu inalterado, variando apenas a origem dos *embeddings* utilizados para definir as similaridades. Como as redes construídas tendem a ser densas, aplicamos um *backbone* adaptativo com limiar definido pelo percentil $p = 80$ dos pesos das arestas, de modo a reduzir a complexidade visual e destacar conexões mais informativas. Para cada rede, foram identificados os jogadores com maior valor nas métricas. Esses vértices foram destacados em Figura 5.11 por meio de símbolos específicos: ★ para *strength* e

◊ para *betweenness*.



(a) Atalanta **(b)** Benevento **(c)** Bologna **(d)** Cagliari

(e) Chievo **(f)** Crotone **(g)** Fiorentina **(h)** Genoa

(i) Hellas Verona **(j)** Internazionale **(k)** Juventus **(l)** Lazio

(m) Milan **(n)** Napoli **(o)** Roma **(p)** Sampdoria

(q) Sassuolo **(r)** SPAL **(s)** Torino **(t)** Udinese

Figura 5.11: Redes de jogadores por clube na Serie A italiana, categorizadas por posição em campo do Modelo BERT.

Nos grafos, os nós foram coloridos de acordo com a posição do jogador em campo. Atacantes foram representados pelos tons mais escuros da escala (roxo), meio-campistas por tons intermediários (azulado), defensores por tons mais claros (esverdeado) e goleiros pelo extremo mais claro (amarelado).

Em termos de padrão, alguns times aparecem com conectividade mais distribuída, com um núcleo bem integrado e várias ligações relevantes entre setores. Por exemplo, a Juventus, em que se observa um miolo denso e a liderança estrutural concentrada no nó destacado, sugerindo um jogador que acumula conexões fortes e também atua como ponte. Em outros, o grafo fica mais centralizado, com poucas peças concentrando as conexões mais intensas e/ou a intermediação, como a Internazionale (4°) e a Udinese (14°), nas quais um defensor destacado em verde sugere papel relevante na circulação e na organização da linha defensiva, e o Napoli (2°), onde o nó dominante em azul indica a centralidade do meio-campo como eixo articulador da equipe. Já em equipes como o Benevento (20°), chama atenção a separação entre os nós marcados pelo grau e centralidade. Isso indica que o jogador com maior soma de conexões (*strength*) não é necessariamente o mesmo que mais costura caminhos mínimos (*betweenness*).

A Tabela 5.2 apresenta, para cada equipe, o jogador mais central segundo duas medidas complementares na rede de similaridade: o grau ponderado, que resume a intensidade total das conexões do nó, e a centralidade de intermediação, que quantifica o quanto o jogador atua como ponte em caminhos mínimos ponderados.

Tabela 5.2: Jogadores centrais por equipe segundo grau ponderado (*strength*) e centralidade de intermediação (*betweenness*) no modelo BERT.

Clube	**Strength**			**Betweenness**		
	Jogador	**Valor**	**Posição**	**Jogador**	**Valor**	**Posição**
Atalanta	Masiello	16,544	Defensor	Palomino	0,158	Defensor
Benevento	Del Pinto	31,055	Meio-campista	Cataldi	0,120	Meio-campista
Bologna	Donsah	24,327	Meio-campista	De Maio	0,153	Defensor
Cagliari	Dessena	16,748	Meio-campista	Rafael	0,130	Goleiro
Chievo	Depaoli	20,060	Meio-campista	Depaoli	0,095	Meio-campista
Crotone	Ceccherini	23,985	Defensor	Capuano	0,162	Defensor
Fiorentina	Benassi	18,707	Meio-campista	Benassi	0,184	Meio-campista
Genoa	Daniel Bessa	18,344	Meio-campista	Daniel Bessa	0,133	Meio-campista
Hellas Verona	Buchel	22,309	Meio-campista	Caracciolo	0,121	Defensor
Internazionale	Nagatomo	18,591	Defensor	Nagatomo	0,246	Defensor
Juventus	Barzagli	28,282	Defensor	Barzagli	0,167	Defensor
Lazio	Lulic	16,887	Meio-campista	Basta	0,175	Defensor
Milan	Bonucci	22,282	Defensor	Bonucci	0,294	Defensor
Napoli	Hamsik	18,376	Meio-campista	Hamsik	0,233	Meio-campista
Roma	Bruno Peres	16,107	Defensor	Bruno Peres	0,225	Defensor
Sampdoria	Barreto	23,642	Meio-campista	Barreto	0,298	Meio-campista
Sassuolo	Peluso	16,417	Defensor	Peluso	0,121	Defensor
SPAL	Mattiello	20,074	Defensor	Mattiello	0,176	Defensor
Torino	Belotti	20,477	Atacante	Belotti	0,316	Atacante
Udinese	Widmer	20,215	Defensor	Widmer	0,268	Defensor

Em termos de resultados, observa-se predominância de defensores e meio-campistas entre os nós centrais, sugerindo que jogadores ligados à construção e à circulação, ou à sustentação defensiva, tendem a concentrar relações mais fortes com múltiplos companheiros, um padrão coerente com o que já havia sido observado nas redes derivadas de GloVe. Da mesma forma, em várias equipes, o mesmo atleta aparece como líder tanto em *strength* quanto em *betweenness* (por exemplo, Atalanta, Chievo, Fiorentina, Genoa, Internazionale, Juventus, Milan, Roma, Sampdoria, Sassuolo, SPAL, Torino e Udinese), reforçando a interpretação, consistente entre os dois modelos, de que alta conectividade costuma caminhar junto de um maior papel de ponte na estrutura do grafo.

Nos poucos casos em que os líderes diferem (como Benevento, Bologna, Cagliari, Crotone, Hellas Verona, Lazio e Napoli), o resultado sugere uma separação mais nítida entre um jogador que concentra similaridades com muitos companheiros (maior *strength*) e outro que atua como elo entre subgrupos (maior *betweenness*), evidenciando a natureza complementar das duas métricas.

A Figura 5.12 apresenta, para a Juventus (campeã) e o Benevento (último colocado), os histogramas das distribuições de grau ponderado (strength) e centralidade de intermediação (betweenness) calculadas a partir dos grafos construídos com BERT.

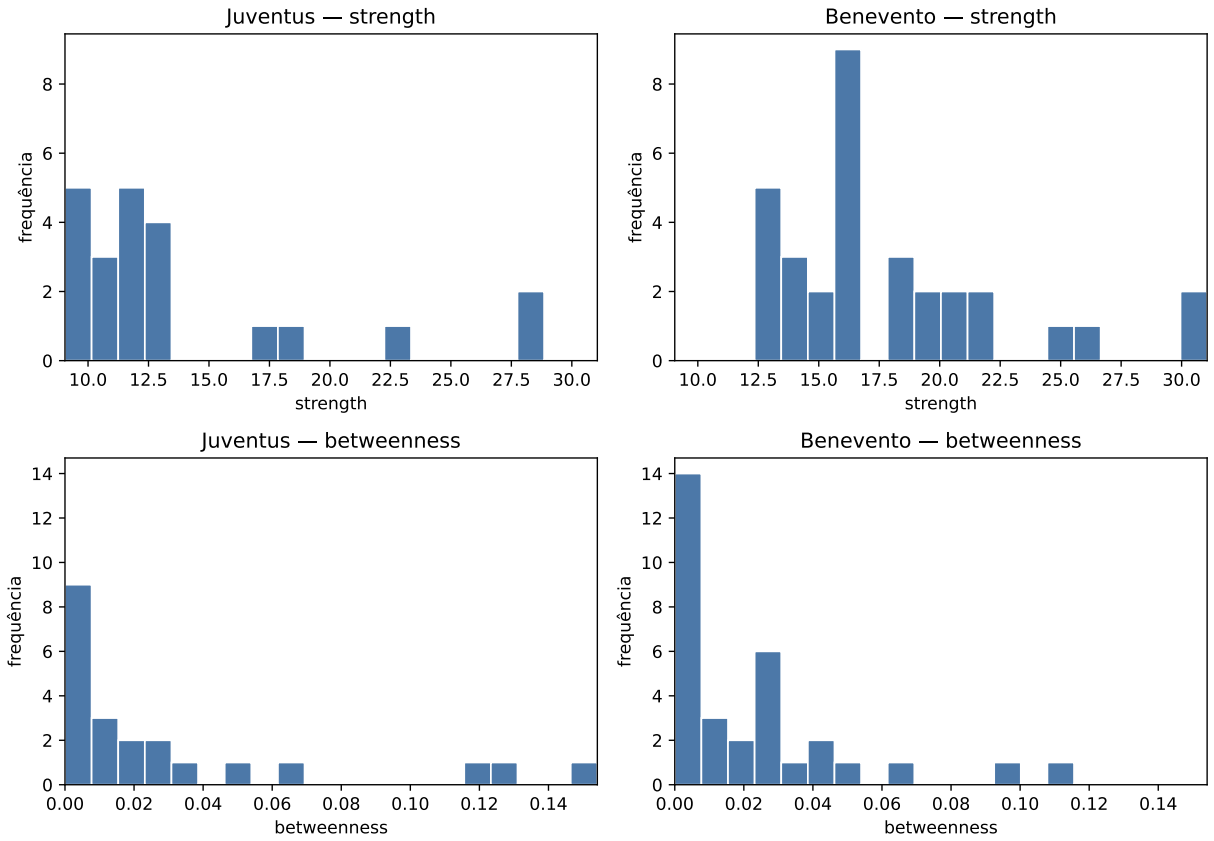


Figura 5.12: Distribuição das métricas de centralidade na Juventus e no Benevento: campeã *vs.* último colocado pelo modelo BERT

Ao comparar as equipes, nota-se que o Benevento apresenta uma distribuição de *strength* mais espalhada e com cauda mais pronunciada, incluindo valores altos, sugerindo uma maior tendência a concentrações de conectividade total em alguns jogadores, ao passo que a Juventus concentra mais jogadores em faixas inferiores e mantém a maior parte com *betweenness* muito baixa, ainda que

existam poucos *outliers*.

Esse padrão é coerente com o que foi discutido no GloVe: equipes de desempenho inferior tendem a exibir maior dependência estrutural de poucos nós, ou seja, maiores valores e maior dispersão das métricas, enquanto equipes mais fortes tendem a apresentar uma organização mais distribuída. No BERT, porém, essa diferença aparece de forma mais sutil, o que é compatível com o efeito de agregação necessário aos *embeddings* contextualizados, reduzindo contraste e homogeneizando parte das similaridades.

NetLSD

A Figura 5.13 apresenta a comparação entre as redes de jogadores das equipes a partir das assinaturas espectrais por meio do NetLSD. No primeiro gráfico, a matriz de distâncias entre equipes é apresentada, enquanto no gráfico à direita, a mesma matriz é reordenada pelo agrupamento hierárquico, acompanhada do dendrograma e do corte utilizado na definição do número de *clusters* (linha tracejada), de modo a evidenciar grupos de maior similaridade estrutural entre redes.

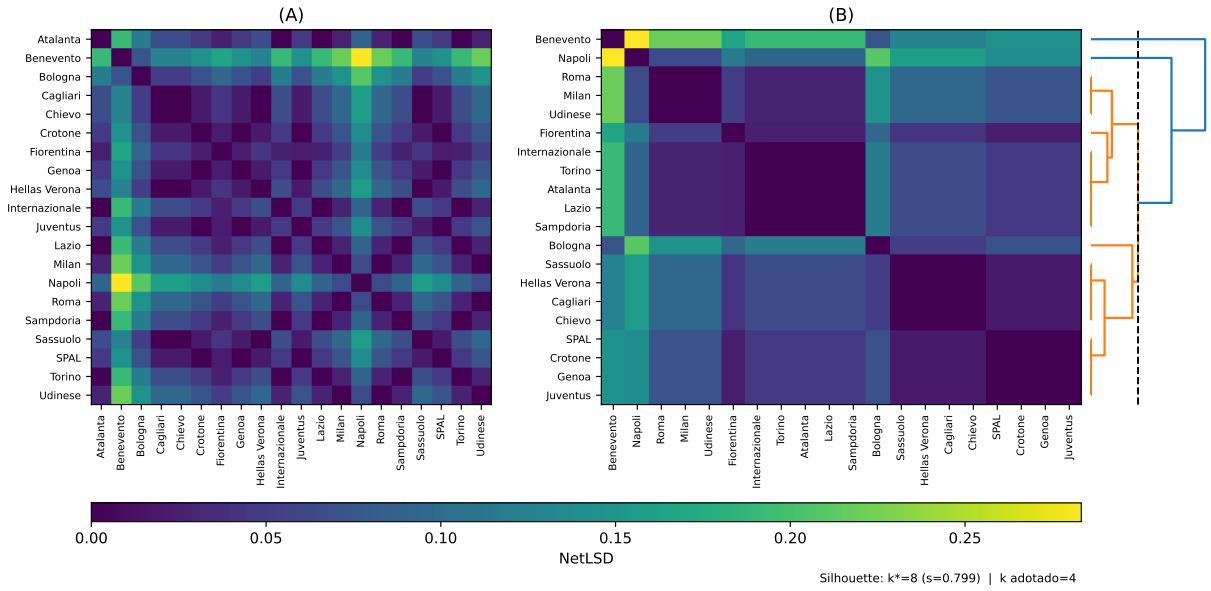


Figura 5.13: (A) Matriz de distâncias e **(B)** agrupamento hierárquico das redes de equipes com base no NetLSD nos *embeddings* do Modelo BERT.

Em termos de leitura, **(A)** é útil como visão de quem está distante de quem no campeonato como um todo, enquanto em **(B)** torna-se mais clara a presença de estruturas em grupos: regiões mais homogêneas indicam conjuntos de times cujas redes apresentam assinaturas espectrais mais próximas entre si, ao passo que zonas com valores mais altos sugerem maior dissimilaridade entre grupos. Além disso, a própria Figura 5.13 reporta a avaliação por *silhouette*, com um $k*$ sugerido e o k adotado para a análise.

Considerando a solução de *clusters* $k = 4$ apresentada, observa-se a formação de dois agrupamentos principais e dois casos isolados. Um agrupamento reúne Atalanta $(7°)$, Sampdoria $(10°)$, Roma $(3°)$, Milan $(6°)$, Lazio $(5°)$, Torino $(9°)$, Internazionale $(4°)$, Fiorentina $(8°)$ e Udinese $(14°)$. Já o

outro agrupamento agrega Genoa ($12°$), Hellas Verona ($19°$), Crotone ($18°$), Juventus ($1°$), Chievo ($13°$), Cagliari ($16°$), Bologna ($15°$), Sassuolo ($11°$) e SPAL ($17°$). Por fim, Napoli ($2°$) aparece como um *cluster* próprio e Benevento ($20°$) como outro agrupamento isolado.

Em comparação com o resultado obtido com GloVe, o agrupamento baseado em BERT mantém a ideia central de que existem "famílias" de equipes com topologias de rede semelhantes, mas tende a produzir uma separação menos contrastante entre os grupos: enquanto no GloVe alguns times apareciam mais nitidamente puxando conglomerados, com distâncias mais marcadas e agrupamentos mais estáveis, aqui a agregação necessária no BERT suaviza as diferenças e faz com que as assinaturas espectrais fiquem mais próximas no geral, resultando em *clusters* mais amplos e com fronteiras menos rígidas. Ainda assim, o padrão de grupos na matriz e o dendrograma indicam que a estrutura relacional capturada pelas redes não é aleatória e que, mesmo com essa suavização, o BERT recupera agrupamentos coerentes, com alguns times se mantendo próximos de vizinhos semelhantes, e *outliers*, como os que formam *clusters* próprios, aparecendo de forma consistente, compatível com o que já havia sido visto no GloVe.

Capítulo 6

Conclusões e trabalhos futuros

Esta dissertação propôs uma metodologia integrada para analisar partidas de futebol a partir de *event data*, combinando *data-to-text generation*, modelos de linguagem e teoria dos grafos. A ideia central foi transformar registros tabulares evento-a-evento em um *corpus* textual padronizado e, a partir dele, obter representações vetoriais que permitissem caracterizar padrões de jogo de forma comparável entre equipes. Na sequência, cada clube foi representado por uma rede ponderada, na qual os nós correspondem a jogadores e as arestas codificam relações induzidas por similaridade de cosseno entre vetores associados aos atletas, possibilitando tanto a identificação de jogadores estruturalmente relevantes quanto a comparação global entre equipes por meio de assinaturas espectrais.

Em termos metodológicos, o fluxo de análise possui três níveis complementares. Primeiro, a etapa textual e vetorial viabilizou inspeções no espaço de vetorial, a visualização da organização das partidas e na interpretação dos *embeddings* foram possíveis com apoio da técnica de redução de dimensionalidade UMAP e de agrupamento HDBScan.

Segundo, a etapa relacional construiu redes por equipe a partir de pesos não negativos (via deslocamento aditivo na similaridade cosseno), o que permitiu aplicar métricas em grafos ponderados. Terceiro, a etapa de comparação via NetLSD, produziu uma matriz de distâncias entre equipes, sobre a qual foi aplicado agrupamento hierárquico, permitindo discutir *clusters* e casos singulares.

Os resultados sustentam a utilidade da abordagem como ferramenta exploratória para comparar assinaturas das equipes. Em particular, a análise baseada em NetLSD sugere agrupamentos entre clubes e também a recorrência de casos isolados, compatíveis com padrões estruturais que se destacam no campeonato. Por exemplo, observam-se dois agrupamentos principais e dois *outliers* que formam *clusters* próprios, sendo que o Benevento aparece como *cluster* isolado em ambas as análises (BERT e GloVe).

Ao comparar os dois modelos de *embeddings* avaliados, a metodologia também evidenciou diferenças interpretáveis: enquanto o *GloVe* tende a produzir separações mais contrastantes, o *BERT* mantém a ideia de famílias de equipes, porém com fronteiras menos rígidas, devido à necessidade de agregação das representações contextualizadas, que suaviza diferenças e aproxima assinaturas no geral. Ainda assim, mesmo com essa suavização, a estrutura encontrada não é aleatória e preserva padrões coerentes no espaço de distâncias, com vizinhanças estáveis e *outliers*..

Por fim, a etapa de visualização das redes mostrou-se importante para interpretação e comunicação

dos achados. Como as redes iniciais tendem a ser densas, foi necessário reduzir complexidade visual por meio de um *backbone* adaptativo por nó, com limiar no percentil $p = 80$, preservando conexões localmente relevantes e mitigando viés em vértices de alto grau. Dentro desse recorte, medidas de centralidade, como *strength* e *betweenness*, forneceram um mecanismo simples para destacar papéis complementares: jogadores com alta conectividade ponderada e jogadores que atuam como pontes estruturais no grafo.

Limitações

Apesar dos resultados obtidos, algumas limitações devem ser consideradas. Em primeiro lugar, **a conversão data-to-text depende de escolhas de notação, vocabulário, granularidade de eventos e segmentação espacial**. Tais decisões influenciam diretamente o *corpus* gerado e, consequentemente, as representações e redes construídas. Além disso, no caso do *BERT*, cada ocorrência de um termo pode produzir um vetor distinto por depender do contexto, o que exige estratégias de armazenamento e, frequentemente, procedimentos de agregação para permitir comparações entre unidades. Essa etapa tende a reduzir o contraste entre equipes, aproximando assinaturas e suavizando separações.

Os **resultados também são sensíveis a hiperparâmetros e decisões de modelagem**. Por exemplo, configurações de UMAP e HDBSCAN, escolha do *backbone* (percentil p), critérios de remoção de vértices isolados, método de ligação no agrupamento hierárquico e definição do número de grupos k, de modo que diferentes combinações podem alterar a estrutura observada. Em particular, o *backbone* por percentis (com $p = 80$) favorece a interpretabilidade, mas modifica a densidade das redes e pode afetar métricas locais.

Outra limitação diz respeito à própria definição das arestas: elas representam a similaridade de cosseno entre vetores associados a jogadores, e não uma interação observada diretamente, como passes entre atletas. Além disso, embora o deslocamento para pesos não negativos viabilize o uso de métricas ponderadas, essa escolha resume relações em um único escalar e pode não capturar aspectos direcionais, temporais ou táticos mais finos. Por fim, os **achados refletem o campeonato e a base de dados utilizados**, de modo que diferenças de liga, temporada, padrão de registro de eventos e mudanças de elenco podem impactar a comparabilidade direta dos resultados.

Trabalhos futuros

Considerando as limitações apresentadas, sugerem-se as seguintes possibilidades de extensão e aprofundamento:

- **Redes dinâmicas e multiescala:** construir grafos por partida, por fase do jogo (*e.g.*, primeiro/segundo tempo) ou por janelas deslizantes, permitindo analisar a estabilidade do estilo e as transições táticas ao longo do tempo, em vez de uma única rede agregada por equipe.
- **Incorporar atributos e direcionalidade:** estender a rede para grafos direcionados e/ou com múltiplas camadas, como, por exemplo, por tipo de evento, zona do campo, contexto de placar,

etc., ou ainda incorporar atributos de nós, como posição, minutos em campo ou contexto de ações, permitindo análises mais ricas do que a similaridade global.

- **Comparadores estruturais adicionais:** comparar *NetLSD* com outros descritores e distâncias entre grafos, por exemplo, métodos baseados em *núcleos de grafos* (*graph kernels*), *transporte ótimo* (*optimal transport*) ou *embeddings* obtidos por *redes neurais em grafos* (*graph neural networks*), avaliando estabilidade, interpretabilidade e custo computacional.
- **Ampliação de escopo e reprodutibilidade:** replicar o estudo em múltiplas temporadas e ligas, avaliando robustez temporal, efeito de mudanças de elenco e sensibilidade a diferentes fornecedores de *event data*.

Em síntese, os resultados obtidos indicam que tratar partidas como sequências textuais e, em seguida, como estruturas relacionais comparáveis oferece um caminho promissor para investigar regularidades de organização coletiva em campo. Ao mesmo tempo, as extensões propostas podem tornar a metodologia mais sensível ao contexto tático, mais estável para comparação entre cenários e mais útil como ferramenta aplicada em análise esportiva.

Referências Bibliográficas

Adjileye, A.A., 2024. Risingballer: A player is a token, a match is a sentence, a path towards a foundational model for football players data analytics. URL: https://arxiv.org/abs/2410.00943, arXiv:2410.00943.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural Machine Translation by Jointly Learning to Align and Translate. URL: http://arxiv.org/abs/1409.0473, doi:10.48550/arXiv.1409.0473. arXiv:1409.0473 version: 1.

Barrat, A., Barthélemy, M., Pastor-Satorras, R., Vespignani, A., 2004. The architecture of complex weighted networks. Proceedings of the National Academy of Sciences 101, 3747–3752. URL: https://www.pnas.org/doi/10.1073/pnas.0400087101, doi:10.1073/pnas.0400087101.

Beltagy, I., Peters, M.E., Cohan, A., 2020. Longformer: The Long-Document Transformer. URL: http://arxiv.org/abs/2004.05150, doi:10.48550/arXiv.2004.05150. arXiv:2004.05150 [cs].

Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., 2003. A neural probabilistic language model. The Journal of Machine Learning Research 3, 1137–1155.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching Word Vectors with Subword Information. URL: http://arxiv.org/abs/1607.04606, doi:10.48550/arXiv.1607.04606. arXiv:1607.04606 [cs] version: 1.

Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L., 1993. The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. 19, 263–311.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language Models are Few-Shot Learners. URL: http://arxiv.org/abs/2005.14165, doi:10.48550/arXiv.2005.14165. arXiv:2005.14165.

Buldú, J.M., Busquets, J., Martínez, J.H., Herrera-Diestra, J.L., Echegoyen, I., Galeano, J., Luque, J., 2018. Using Network Science to Analyse Football Passing Networks: Dynamics, Space, Time, and the Multilayer Nature of the Game. Frontiers in Psychology 9. URL: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2018.01900/full, doi:10.3389/fpsyg.2018.01900.

Campello, R.J.G.B., Moulavi, D., Sander, J., 2013. Density-Based Clustering Based on Hierarchical Density Estimates, in: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (Eds.), Advances in Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg. pp. 160–172. doi:10.1007/978-3-642-37456-2_14.

Capellera, G., Ferraz, L., Rubio, A., Agudo, A., Moreno-Noguer, F., 2024. Footbots: A transformer-based architecture for motion prediction in soccer, pp. 2313–2319. doi:10.1109/ICIP51287.2024.10647396.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N., 2022. PaLM: Scaling Language Modeling with Pathways. URL: http://arxiv.org/abs/2204.02311, doi:10.48550/arXiv.2204.02311. arXiv:2204.02311.

Chung, F.R.K., 1997. Spectral Graph Theory. American Mathematical Soc. Google-Books-ID: 4IK8DgAAQBAJ.

Church, K.W., Mercer, R.L., 1993. Introduction to the Special Issue on Computational Linguistics Using Large Corpora. Comput. Linguistics URL: https://www.semanticscholar.org/paper/Introduction-to-the-Special-Issue-on-Computational-Church-Mercer/95690ffe97521a7f95884d095458fbcf2bf09c84.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297. URL: http://link.springer.com/10.1007/BF00994018, doi:10.1007/BF00994018.

Cullingford, R., 1977. SAM: a program that uses world knowledge to understand. ACM SIGART Bulletin , 53–54URL: https://dl.acm.org/doi/10.1145/1045283.1045324, doi:10.1145/1045283.1045324.

Deloitte's Sports Business Group, 2023. Annual Review of Football Finance. URL: https://www2.deloitte.com/uk/en/pages/sports-business-group/articles/annual-review-of-football-finance-europe.html.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Burstein, J., Doran, C., Solorio, T. (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. URL: https://aclanthology.org/N19-1423, doi:10.18653/v1/N19-1423.

Diestel, R., 2017. Graph Theory: 5th edition. Google-Books-ID: zIxRDwAAQBAJ.

Fernandes, P., Nunes, S., Santos, L., 2024. A Community-Driven Data-to-Text Platform for Football Match Summaries, in: Calzolari, N., Kan, M.Y., Hoste, V., Lenci, A., Sakti, S., Xue, N. (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia. pp. 164–173. URL: https://aclanthology.org/2024.lrec-main.15/.

Fernández, J., 2021. A framework for the analytical and visual interpretation of complex spatiotemporal dynamics in soccer. Ph.D. thesis. Polytechnic University of Catalonia.

Freeman, L.C., 1977. A Set of Measures of Centrality Based on Betweenness. Sociometry 40, 35–41. URL: https://www.jstor.org/stable/3033543, doi:10.2307/3033543.

Fruchterman, T.M.J., Reingold, E.M., 1991. Graph drawing by force-directed placement. Software: Practice and Experience 21, 1129–1164. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102, doi:10.1002/spe.4380211102. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.4380211102.

Gatt, A., Krahmer, E., 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. URL: http://arxiv.org/abs/1703.09902, doi:10.48550/arXiv.1703.09902. arXiv:1703.09902 [cs].

Horton, M., Lucey, P., 2025. Large-scale in-game outcome forecasting for match, team and players in football using an axial transformer neural network. URL: https://arxiv.org/abs/2511.18730, arXiv:2511.18730.

Hutchins, W.J., 1995. Machine Translation: A Brief History, in: Koerner, E.F.K., Asher, R.E. (Eds.), Concise History of the Language Sciences. Pergamon, Amsterdam, pp. 431–445. URL: https://www.sciencedirect.com/science/article/pii/B9780080425801500660, doi:10.1016/B978-0-08-042580-1.50066-0.

Jelinek, F., 1990. SELF-ORGANIZED LANGUAGE MODELING FOR SPEECH RECOGNITION, Elsevier. pp. 450–506. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780080515847500450, doi:10.1016/B978-0-08-051584-7.50045-0. book Title: Readings in Speech Recognition.

Kumar, E., 2013. Natural Language Processing. I. K. International Pvt Ltd. Google-Books-ID: FpUBFNFuKWgC.

Lane, H., Hapke, H., Howard, C., 2019. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python. First edition ed., Manning, Shelter Island, NY.

Li, B., Han, L., 2013. Distance Weighted Cosine Similarity Measure for Text Classification, in: Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., Yao, X. (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2013, Springer, Berlin, Heidelberg. pp. 611–618. doi:10.1007/978-3-642-41278-3_74.

Li, Y., Ma, R., Gonçalves, B., Gong, B., Cui, Y., Shen, Y., 2020. Data-driven team ranking and match performance analysis in Chinese Football Super League. Chaos, Solitons & Fractals 141, 110330. URL: https://www.sciencedirect.com/science/article/pii/S0960077920307256, doi:10.1016/j.chaos.2020.110330.

Liu, S., Bremer, P.T., Thiagarajan, J.J., Srikumar, V., Wang, B., Livnat, Y., Pascucci, V., 2018. Visual exploration of semantic relationships in neural word embeddings. IEEE Transactions on Visualization and Computer Graphics 24. doi:10.1109/TVCG.2017.2745141.

Liu, T., Zheng, X., Chang, B., Sui, Z., 2021. Towards Faithfulness in Open Domain Table-to-text Generation from an Entity-centric View. URL: http://arxiv.org/abs/2102.08585, doi:10.48550/arXiv.2102.08585. arXiv:2102.08585 [cs].

Léon, J., 2014. Early Machine Translation, in: Beckmann, A., Csuhaj-Varjú, E., Meer, K. (Eds.), Language, Life, Limits, Springer International Publishing, Cham. pp. 275–282. doi:10.1007/978-3-319-08019-2_28.

McInnes, L., Healy, J., Melville, J., 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. URL: https://arxiv.org/abs/1802.03426v2.

Mendes-Neves, T., Meireles, L., Mendes-Moreira, J., 2024a. Forecasting Events in Soccer Matches Through Language. URL: https://arxiv.org/abs/2402.06820v2.

Mendes-Neves, T., Meireles, L., Mendes-Moreira, J., 2024b. Towards a foundation large events model for soccer. Machine Learning 113, 8687–8709. URL: https://doi.org/10.1007/s10994-024-06606-y, doi:10.1007/s10994-024-06606-y.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. URL: http://arxiv.org/abs/1301.3781, doi:10.48550/arXiv.1301.3781. arXiv:1301.3781 [cs].

Miller, G.A., 1995. WordNet: a lexical database for English. Commun. ACM 38, 39–41. URL: https://dl.acm.org/doi/10.1145/219717.219748, doi:10.1145/219717.219748.

Moryossef, A., Goldberg, Y., Dagan, I., 2019. Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation. URL: http://arxiv.org/abs/1904.03396, doi:10.48550/arXiv.1904.03396. arXiv:1904.03396 [cs].

Müller, D.W., Sawitzki, G., 1991. Excess Mass Estimates and Tests for Multimodality. Journal of the American Statistical Association 86, 738–746. URL: https://www.jstor.org/stable/2290406, doi:10.2307/2290406.

Newman, M., 2010. Networks: An Introduction. OUP Oxford. Google-Books-ID: LrFaU4XCsUoC.

Oliveira, J.R.d., Tobar, F.B., Capraro, A.M., 2021. Football tourism: a bibliometric analysis of published works in the tourism-based journals (2003–2019). Journal of Sport & Tourism 25, 317–335. URL: https://doi.org/10.1080/14775085.2021.1965010, doi:10.1080/14775085.2021.1965010. _eprint: https://doi.org/10.1080/14775085.2021.1965010.

Pal, S., Chang, M., Iriarte, M.F., 2022. Summary Generation Using Natural Language Processing Techniques and Cosine Similarity, in: Abraham, A., Gandhi, N., Hanne, T., Hong, T.P., Nogueira Rios, T., Ding, W. (Eds.), Intelligent Systems Design and Applications, Springer International Publishing, Cham. pp. 508–517. doi:10.1007/978-3-030-96308-8_47.

Pantzalis, V.C., Tjortjis, C., 2020. Sports Analytics for Football League Table and Player Performance Prediction, in: 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA, IEEE, Piraeus, Greece. pp. 1–8. URL: https://ieeexplore.ieee.org/document/9284352/, doi:10.1109/IISA50023.2020.9284352.

Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., Giannotti, F., 2019. A public data set of spatio-temporal match events in soccer competitions. Scientific Data 6, 236. URL: https://www.nature.com/articles/s41597-019-0247-7, doi:10.1038/s41597-019-0247-7.

Park, K., Hong, J.S., Kim, W., 2020. A Methodology Combining Cosine Similarity with Classifier for Text Classification. Applied Artificial Intelligence 34, 396–411. URL: https://doi.org/10.1080/08839514.2020.1723868, doi:10.1080/08839514.2020.1723868. _eprint: https://doi.org/10.1080/08839514.2020.1723868.

Pennington, J., Socher, R., Manning, C., 2014. GloVe: Global Vectors for Word Representation, in: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar. pp. 1532–1543. URL: https://aclanthology.org/D14-1162, doi:10.3115/v1/D14-1162.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. URL: http://arxiv.org/abs/1802.05365, doi:10.48550/arXiv.1802.05365. arXiv:1802.05365 [cs].

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. URL: http://arxiv.org/abs/1910.10683, doi:10.48550/arXiv.1910.10683. arXiv:1910.10683.

Reiter, E., Dale, R., 2000. Building Natural Language Generation Systems. Studies in Natural Language Processing, Cambridge University Press, Cambridge. URL: https://www.cambridge.org/core/books/building-natural-language-generation-systems/0AE70C709A9BFBDC80B349B2D22A78CD, doi:10.1017/CBO9780511519857.

Rogers, A., Kovaleva, O., Rumshisky, A., 2021. A Primer in BERTology: What We Know About How BERT Works. Transactions of the Association for Computational Linguistics 8, 842–866. URL: https://doi.org/10.1162/tacl_a_00349, doi:10.1162/tacl_a_00349.

Romero, R., Mashayekhi, Y., Lai, F., Van Roy, M., De Bie, T., Davis, J., 2026. Next-Event Prediction in Soccer: Assessing the Impact of Team and Player Information, in: Rios-Neto, H., Robberechts, P., Van Roy, M., Zimmermann, A. (Eds.), Machine Learning and Data Mining for Sports Analytics, Springer Nature Switzerland, Cham. pp. 62–73. doi:10.1007/978-3-032-15165-0_5.

Sak, A., 2023. Using cosine similarity classifier for NLP analysis in construction field texts. AIP Conference Proceedings 2791, 040010. URL: https://doi.org/10.1063/5.0143607, doi:10.1063/5.0143607.

Schank, R.C., 1975. Conceptual Information Processing. North-Holland. Google-Books-ID: GYdQAAAAMAAJ.

Serrano, M., Boguñá, M., Vespignani, A., 2009. Extracting the multiscale backbone of complex weighted networks. Proceedings of the National Academy of Sciences 106, 6483–6488. URL: https://www.pnas.org/doi/10.1073/pnas.0808904106, doi:10.1073/pnas.0808904106.

Singhal, A., 2001. Modern Information Retrieval: A Brief Overview 24, 35–43. URL: http://sites.computer.org/debull/A01DEC-CD.pdf.

Spearman, W., 2018. Beyond expected goals. MIT Sloan Sports Analytics Conference .

Sun, C., Qiu, X., Xu, Y., Huang, X., 2019. How to Fine-Tune BERT for Text Classification?, in: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (Eds.), Chinese Computational Linguistics, Springer International Publishing, Cham. pp. 194–206. doi:10.1007/978-3-030-32381-3_16.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to Sequence Learning with Neural Networks. URL: http://arxiv.org/abs/1409.3215, doi:10.48550/arXiv.1409.3215. arXiv:1409.3215.

Taniguchi, Y., Feng, Y., Takamura, H., Okumura, M., 2019. Generating live soccer-match commentary from play data, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI Press, Honolulu, Hawaii, USA. pp. 7096–7103. URL: https://dl.acm.org/doi/10.1609/aaai.v33i01.33017096, doi:10.1609/aaai.v33i01.33017096.

Taylor, W.L., 1953. "Cloze Procedure": A New Tool for Measuring Readability. Journalism Quarterly 30, 415–433. URL: https://journals.sagepub.com/doi/10.1177/107769905303000401, doi:10.1177/107769905303000401.

Teixeira, J.E., Leal, M., Ferraz, R., Ribeiro, J., Cachada, J.M., Barbosa, T.M., Monteiro, A.M., Forte, P., 2021. Effects of Match Location, Quality of Opposition and Match Outcome on Match Running Performance in a Portuguese Professional Football Team. Entropy 23, 973. URL: https://www.mdpi.com/1099-4300/23/8/973, doi:10.3390/e23080973. number: 8.

Tobar, F., Ramshaw, G., 2022. "Welcome to the EPL": analysing the development of football tourism in the English Premier League. Soccer & Society 23, 432–450. URL: https://doi.org/10.1080/14660970.2022.2059871, doi:10.1080/14660970.2022.2059871. _eprint: https://doi.org/10.1080/14660970.2022.2059871.

Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A., Müller, E., 2018. NetLSD: Hearing the Shape of a Graph, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2347–2356. URL: http://arxiv.org/abs/1805.10712, doi:10.1145/3219819.3219991. arXiv:1805.10712 [cs].

Tumminello, M., Micciché, S., Lillo, F., Piilo, J., Mantegna, R.N., 2011. Statistically Validated Networks in Bipartite Complex Systems. PLOS ONE 6, e17994. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0017994, doi:10.1371/journal.pone.0017994.

Turing, A., 1950. Computing Machinery and Intelligence. Mind 59, 433–460. URL: https://doi.org/10.1093/oso/9780198250791.003.0017, doi:10.1093/oso/9780198250791.003.0017.

Vajjala, S., Majumder, B., Gupta, A., Surana, H., 2020. Practical Natural Language Processing: A Comprehensive Guide to Building Real-world NLP Systems. O'Reilly Media. Google-Books-ID: G40jywEACAAJ.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention Is All You Need. URL: http://arxiv.org/abs/1706.03762, doi:10.48550/arXiv.1706.03762. arXiv:1706.03762 [cs] version: 1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2023. Attention Is All You Need. URL: http://arxiv.org/abs/1706.03762, doi:10.48550/arXiv.1706.03762. arXiv:1706.03762 [cs].

Weizenbaum, J., 1966. ELIZA—a computer program for the study of natural language communication between man and machine. Commun. ACM 9, 36–45. URL: https://dl.acm.org/doi/10.1145/365153.365168, doi:10.1145/365153.365168.

West, D.B., 1996. Introduction to Graph Theory. Prentice Hall. Google-Books-ID: HuvuAAAAMAAJ.

Winograd, T., 1972. Understanding natural language. Cognitive Psychology 3, 1–191. URL: https://www.sciencedirect.com/science/article/pii/0010028572900023, doi:10.1016/0010-0285(72)90002-3.

Wiseman, S., Shieber, S.M., Rush, A.M., 2017. Challenges in Data-to-Document Generation. URL: http://arxiv.org/abs/1707.08052, doi:10.48550/arXiv.1707.08052. arXiv:1707.08052 [cs].

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A., 2020. Transformers: State-of-the-Art Natural Language Processing, in: Liu, Q., Schlangen, D. (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online. pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6/, doi:10.18653/v1/2020.emnlp-demos.6.

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, , Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. URL: http://arxiv.org/abs/1609.08144, doi:10.48550/arXiv.1609.08144. arXiv:1609.08144.

Yi, Q., Groom, R., Dai, C., Liu, H., Gómez Ruano, M., 2019. Differences in Technical Performance of Players From 'The Big Five' European Football Leagues in the UEFA Champions League. Frontiers in Psychology 10. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2019.02738.

Yi, Q., Gómez-Ruano, M., Liu, H., Zhang, S., Gao, B., Wunderlich, F., Memmert, D., 2020. Evaluation of the Technical Performance of Football Players in the UEFA Champions League. International Journal of Environmental Research and Public Health 17, 604. URL: https://www.mdpi.com/1660-4601/17/2/604, doi:10.3390/ijerph17020604. number: 2.

Zhang, W., Yang, Y., 2025. Fot: an efficient transformer framework for real-time small object detection in football videos. Scientific Reports 15, 30875.

Anexo A

Zonas da segmentação do campo

As zonas foram nomeadas em inglês para manter consistência com a implementação. Na notação adotada, os prefixos *D* e *O* indicam, respectivamente, as metades defensiva (*defensive*) e ofensiva (*offensive*) do campo; *L* e *R* referem-se aos lados esquerdo (*left*) e direito (*right*); *CZ* denota a zona central (*central zone*); *LLZ* e *RLZ* representam as zonas laterais esquerda e direita (*left/right lateral zone*); *SAZ* corresponde à pequena área (*six-yard area zone*) e *LAZ* à grande área (*penalty area zone*).

- **Zona da Pequena Área Defensiva** — *Defensive Six-Yard Area Zone* (DSAZ)
- **Zona da Grande Área Defensiva** — *Defensive Penalty Area Zone* (DLAZ)
- **Zona Lateral Esquerda Defensiva** — *Defensive Left Lateral Zone* (DLLZ)
- **Zona Lateral Direita Defensiva** — *Defensive Right Lateral Zone* (DRLZ)
- **Zona Central do Meio Defensiva** — *Mid Defensive Central Zone* (MDCZ)
- **Zona Central Esquerda Defensiva** — *Defensive Left Central Zone* (DLCZ)
- **Zona Central Direita Defensiva** — *Defensive Right Central Zone* (DRCZ)
- **Zona Central Defensiva** — *Defensive Central Zone* (DCZ)
- **Zona Central Esquerda Ofensiva** — *Offensive Left Central Zone* (OLCZ)
- **Zona Central Direita Ofensiva** — *Offensive Right Central Zone* (OCDZ)
- **Zona Central Ofensiva** — *Offensive Central Zone* (OCZ)
- **Zona Lateral Esquerda Ofensiva** — *Offensive Left Lateral Zone* (OLLZ)
- **Zona Lateral Direita Ofensiva** — *Offensive Right Lateral Zone* (ORLZ)
- **Zona Central do Meio Ofensiva** — *Mid Offensive Central Zone* (MOCZ)
- **Zona da Pequena Área Ofensiva** — *Offensive Six-Yard Area Zone* (OSAZ)
- **Zona da Grande Área Ofensiva** — *Offensive Penalty Area Zone* (OLAZ)

Anexo B

Distribuição das métricas de grau e centralidade, GloVe

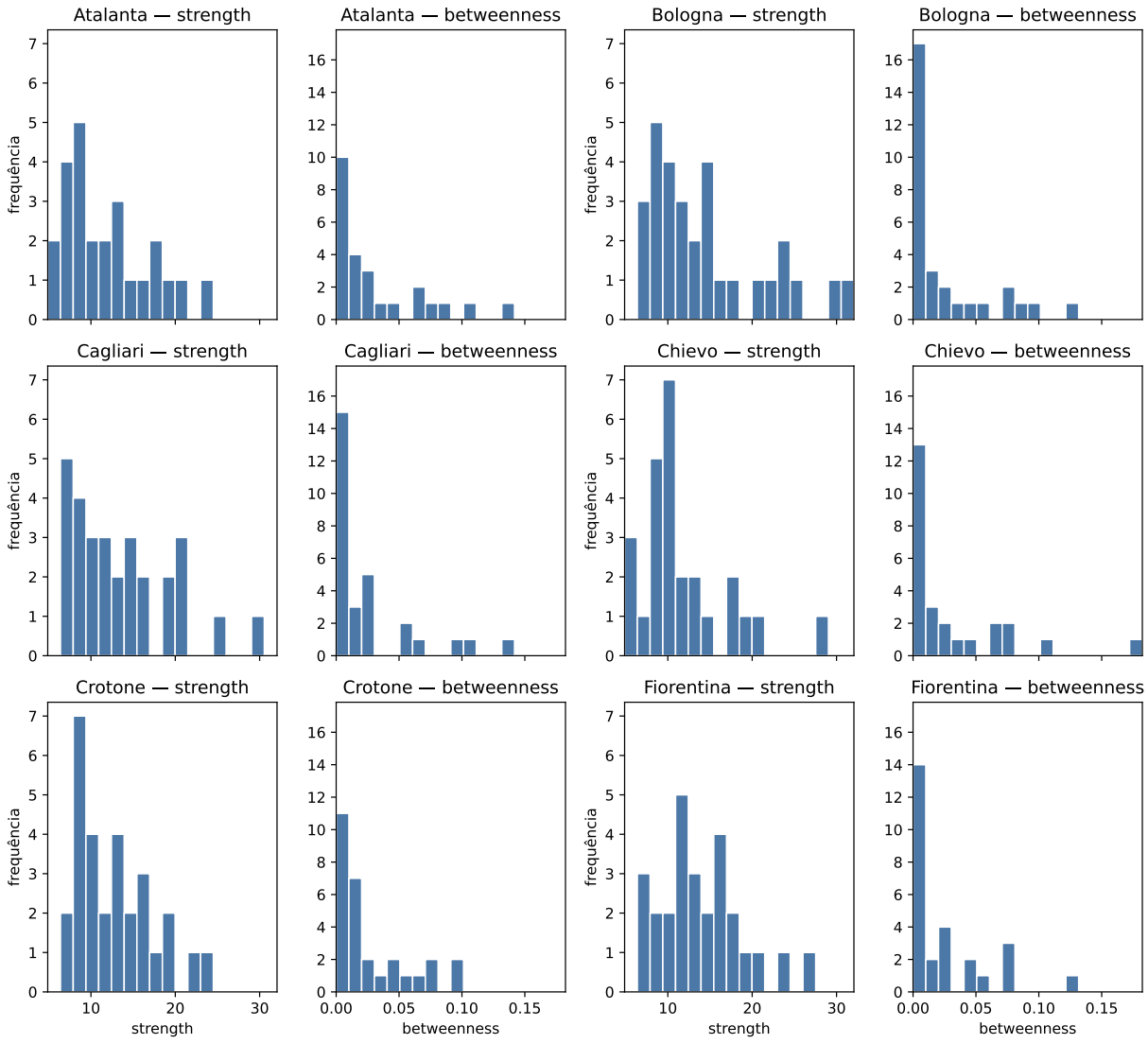


Figura B.1: Distribuições de *strength* e *betweenness* dos jogadores em seis equipes (GloVe): Atalanta, Bologna, Cagliari, Chievo, Crotone e Fiorentina.

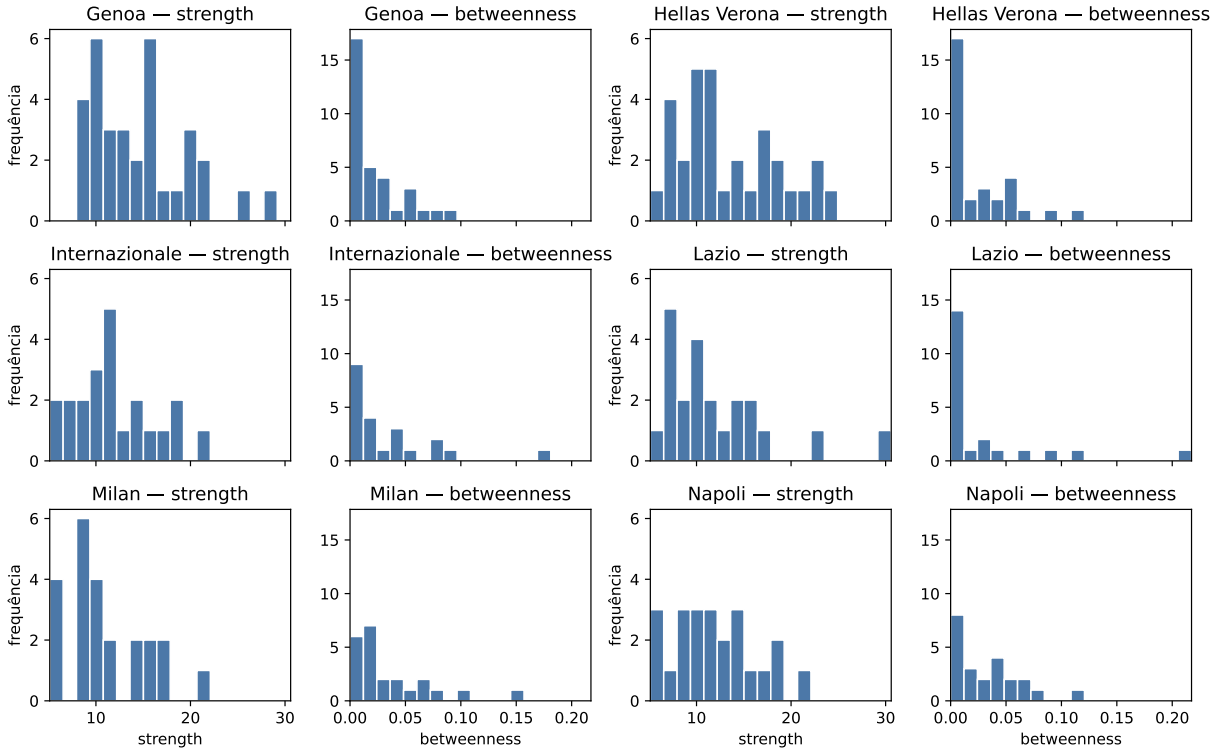


Figura B.2: Distribuições de *strength* e *betweenness* dos jogadores em seis equipes (GloVe): Genoa, Hellas Verona, Internazionale, Lazio, Milan e Napoli.

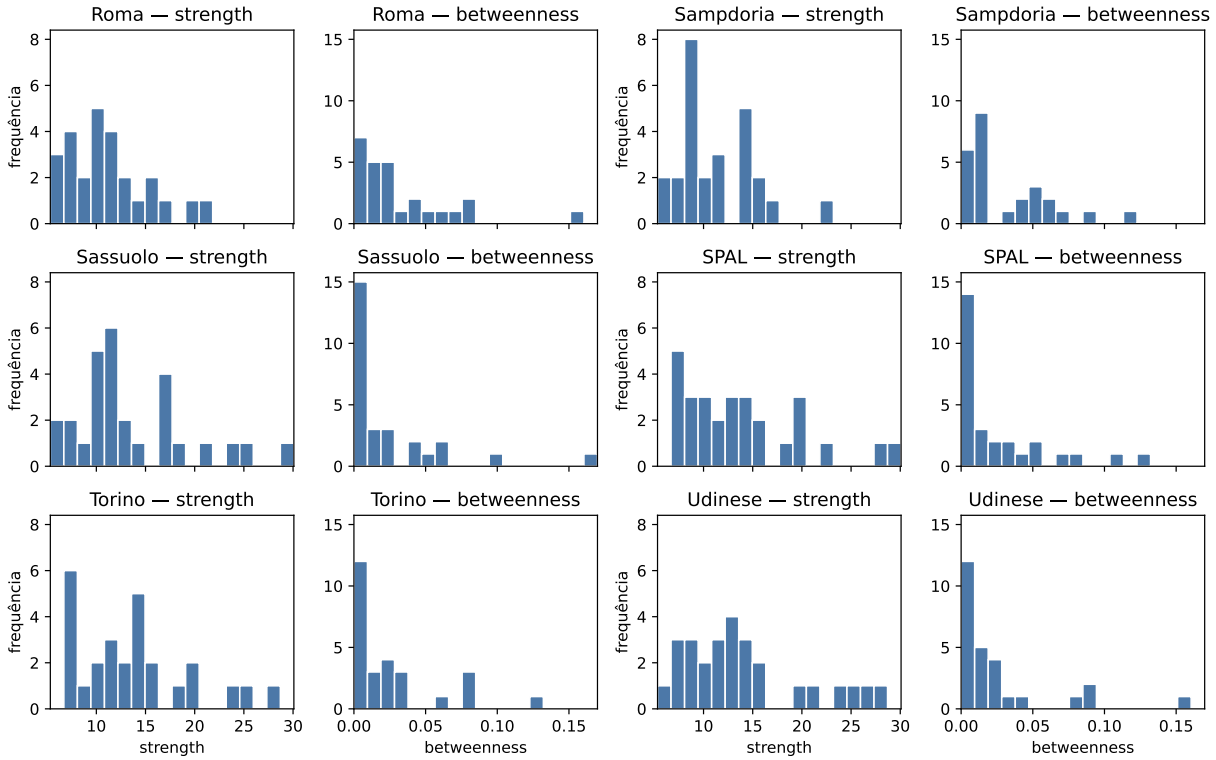


Figura B.3: Distribuições de *strength* e *betweenness* dos jogadores em seis equipes (GloVe): Roma, Sampdoria, Sassuolo, SPAL, Torino e Udinese.

Anexo C

Distribuição das métricas de grau e centralidade, BERT

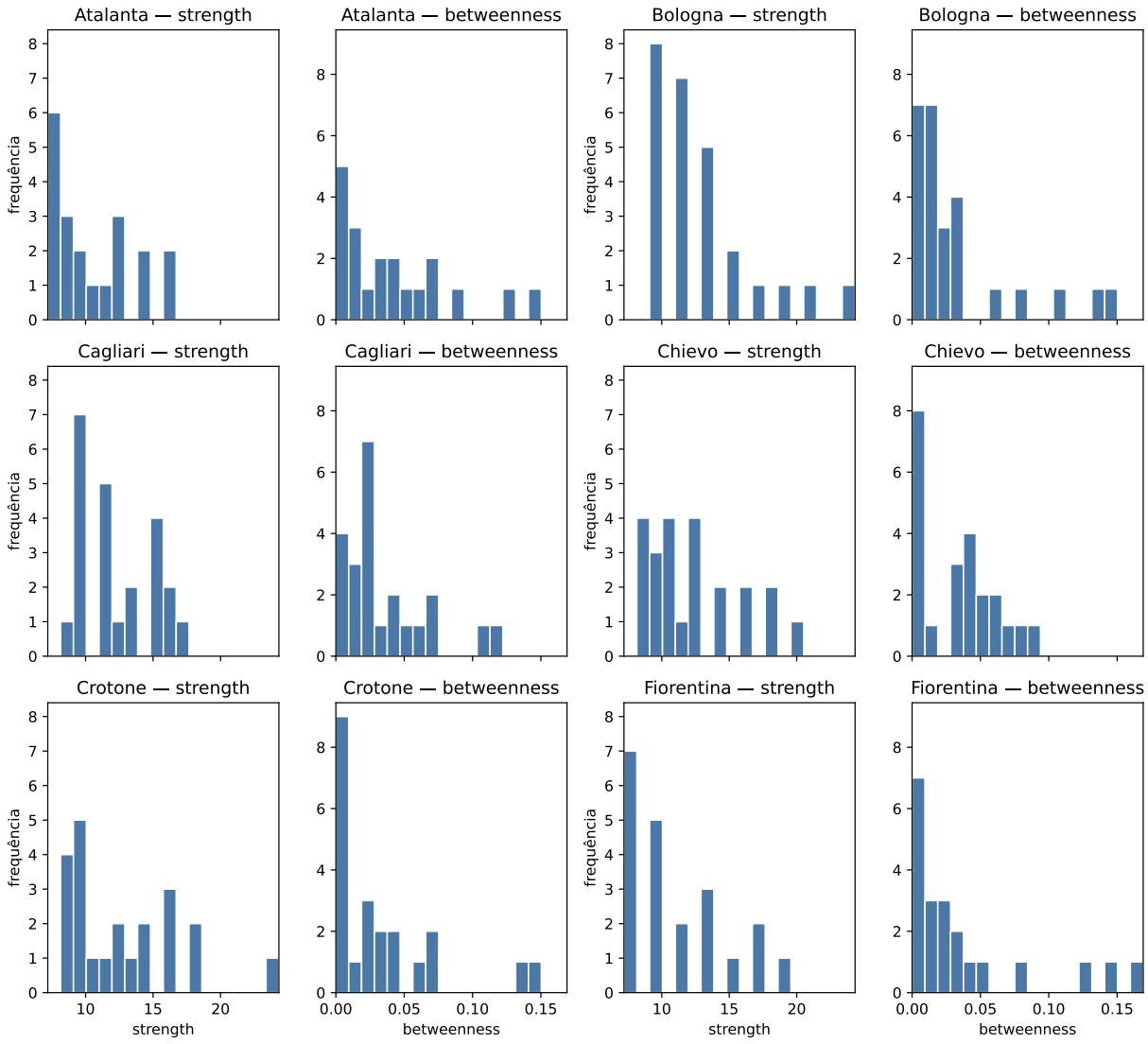


Figura C.1: Distribuições de *strength* e *betweenness* dos jogadores em seis equipes (BERT): Atalanta, Bologna, Cagliari, Chievo, Crotone e Fiorentina.

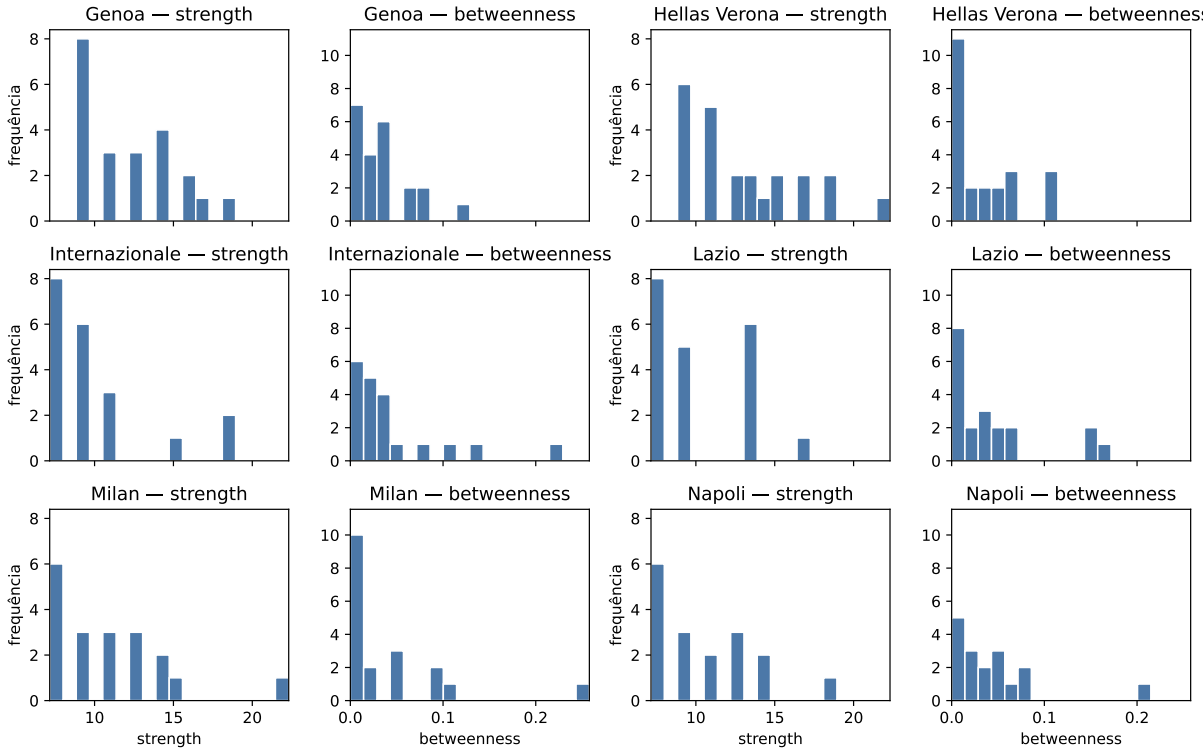


Figura C.2: Distribuições de *strength* e *betweenness* dos jogadores em seis equipes (BERT): Genoa, Hellas Verona, Internazionale, Lazio, Milan e Napoli.

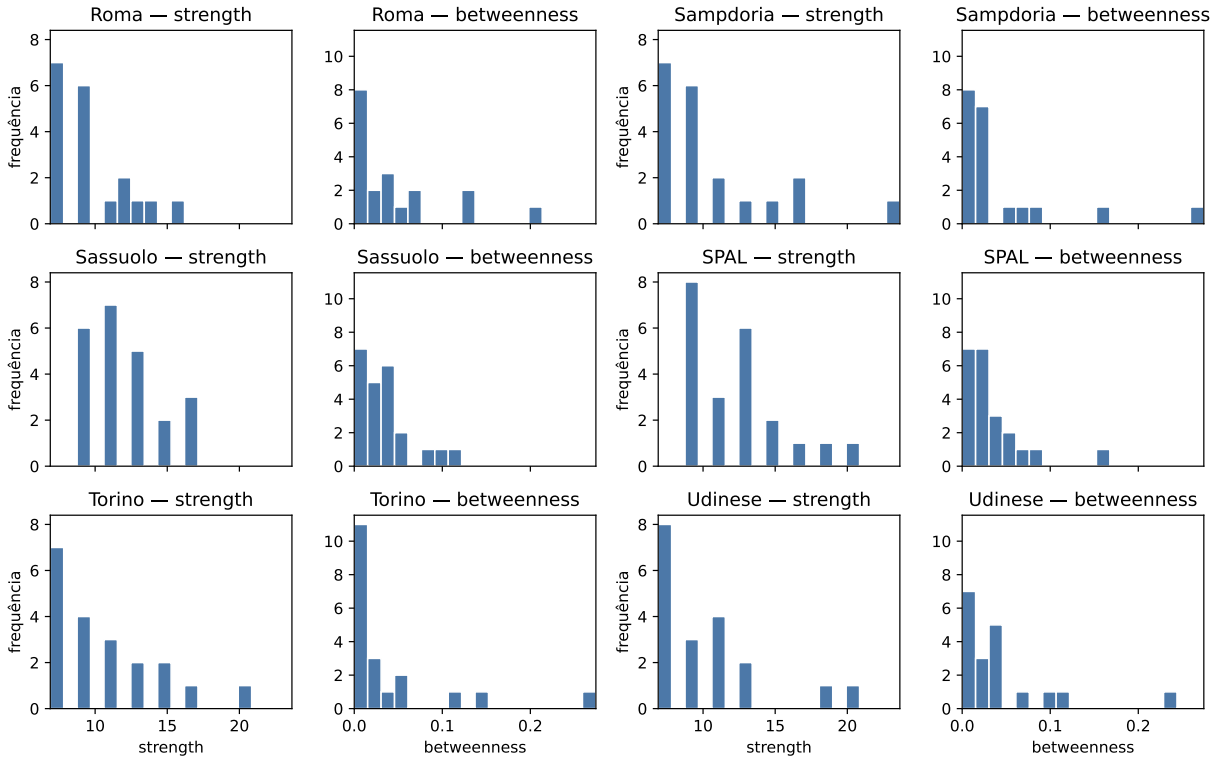


Figura C.3: Distribuições de *strength* e *betweenness* dos jogadores em seis equipes (BERT): Roma, Sampdoria, Sassuolo, SPAL, Torino e Udinese.

Anexo D

Escolha de k para o agrupamento das equipes, GloVe

A Figura D.1 apresenta o comportamento do índice de *silhouette* médio para diferentes valores de k, utilizado como critério auxiliar para orientar a escolha do número de grupos no agrupamento das equipes baseado em distâncias NetLSD.

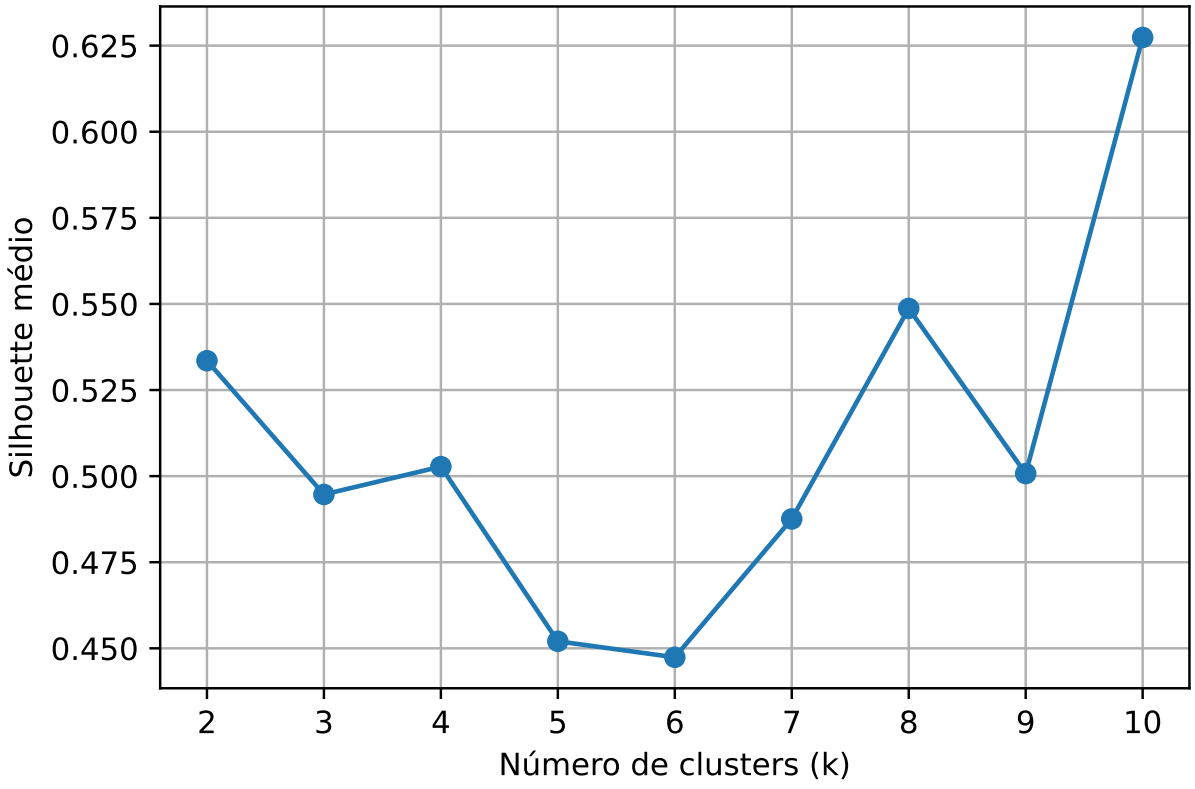


Figura D.1: Seleção do número de *clusters*: *silhouette* médio para a matriz de distâncias NetLSD (GloVe).

Anexo E

Escolha de k para o agrupamento das equipes, BERT

A Figura E.1 apresenta o comportamento do índice de *silhouette* médio para diferentes valores de k, utilizado como critério auxiliar para orientar a escolha do número de grupos no agrupamento das equipes baseado em distâncias NetLSD.

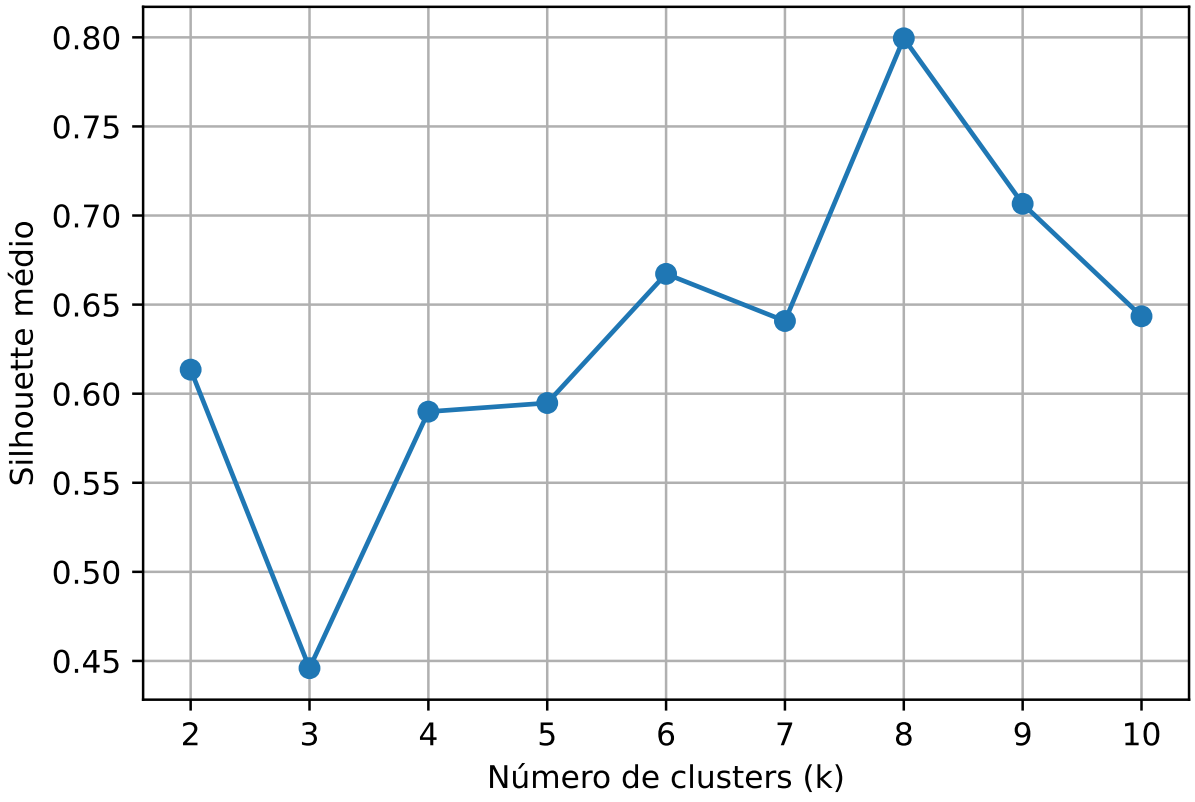


Figura E.1: Seleção do número de *clusters*: *silhouette* médio para a matriz de distâncias NetLSD (BERT)