UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Eduardo Vinicius Betim

Análise Do Comportamento De Comunidade Sobre Dados Da Guerra Da Ucrânia No Reddit

Florianópolis
2024

Eduardo Vinicius Betim

Análise Do Comportamento De Comunidade Sobre Dados Da Guerra Da Ucrânia No Reddit

Trabalho de Conclusão de Curso do Curso de Graduação em Ciência da Computação do Campus Florianópolis da Universidade Federal de Santa Catarina para a obtenção do título de bacharel em Ciência da Computação.
Orientadora: Prof[a]. Carina Friedrich Dorneles, Dr[a].
Coorientador: Prof. Eric Fernandes de Mello Araújo, Dr. (Universidade Federal de Lavras)

Florianópolis
2024

Eduardo Vinicius Betim

Análise Do Comportamento De Comunidade Sobre Dados Da Guerra Da Ucrânia No Reddit

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "bacharel em Ciência da Computação" e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação.

Florianópolis, 26 de Junho de 2024.

Profª. Lúcia Helena Martins Pacheco, Dra.
Coordenadora do Curso

Banca Examinadora:

Profª. Carina Friedrich Dorneles, Drª.
Orientadora

Prof. Eric Fernandes de Mello Araujo, Dr.
Coorientador
Universidade Federal de Lavras

Profª. Ana Paula Couto da Silva, Drª.
Avaliadora
Universidade Federal de Minas Gerais

Profª. Mirella Moura Moro, Drª.
Avaliadora
Universidade Federal de Minas Gerais

AGRADECIMENTOS

Eu declaro meus mais profundos agradecimentos a todos que fizeram parte de minha jornada até a elaboração deste trabalho. Foi o suporte de muitas pessoas que me manteve motivado para tentar dar o melhor de mim todo dia.

Agradeço minha orientadora Carina e meu coorientador Eric por auxiliarem no processo de desenvolvimento deste trabalho com ideias, explicações e opiniões que foram de suma importância para o resultado final.

Agradeço a minha mãe, Simone, e minha avó, Adelia, por serem os pilares que me sustentaram desde o início, com todo seu carinho, compreensão e amor.

Agradeço a minha namorada, Júlia, por ter estado ao meu lado em todos os momentos, bons e ruins, por sempre acreditar no meu potencial e enfrentar minhas dificuldades junto comigo.

Agradeço ao Alisson, por ser minha dupla dentro da graduação e um grande amigo fora dela. Todas as conversas e as risadas que dividimos tornaram essa jornada muito mais fácil e leve.

Agradeço aos meus amigos Ariel, Bruno, Caio, Gabriel, José, Anderson, Marcos, Silvia, Leonardo, Antônio, Victor, Alexandre e Pedro por todos os momentos juntos que tornaram meus dias mais divertidos.

E agradeço também a Universidade Federal de Santa Catarina e todos os docentes que participaram da minha trajetória e contribuíram para meu desenvolvimento profissional e pessoal.

RESUMO

Por meio das redes sociais, consegue-se obter de forma instantânea informações sobre qualquer assunto que desejamos, observando os tópicos de discussão que surgem espontaneamente a partir das interações entre pessoas no ambiente virtual. Assim como no mundo real, dentro das mídias sociais as pessoas tendem a convergir ou divergir baseado em seus interesses e opiniões pessoais, formando grupos que possuem características e pontos de vista particulares. É interessante analisar estes grupos, de forma a compreender melhor o propósito dessas plataformas na sociedade, bem como o comportamento das pessoas que as utilizam, que expõe os pontos de vista e cultura presentes. Este trabalho tem como princípio estudar estes círculos sociais como meio de identificar conjuntos de indivíduos similares, opiniões comuns e mudanças nas interações sociais com base em um evento de grande impacto mundial, utilizando algoritmos de detecção de comunidade em dados extraídos de comentários e postagens da rede social Reddit sobre a guerra entre Rússia e Ucrânia entre os anos de 2022 e 2023. Inicialmente, os dados passam por um processo de caracterização para identificar suas propriedades relevantes, para então servirem de auxílio no objetivo final deste trabalho - modelar grafos que permitirão a observação e análise mais aprofundada das comunidades presentes no material investigado.

Palavras-chave: Detecção de Comunidades. Redes de opinião. Reddit. Guerra Russo-Ucraniana. Caracterização de dados.

ABSTRACT

Through social networks, it is possible to instantly obtain information on any subject we desire by observing the discussion topics that spontaneously arise from interactions between people in the virtual environment. Just like in the real world, within social media, people tend to converge or diverge based on their interests and personal opinions, forming groups that have particular characteristics and viewpoints. It is interesting to analyze these groups to better understand the purpose of these platforms in society, as well as the behavior of the people who use them, which reveals the present viewpoints and culture. This work aims to study these social circles as a means of identifying sets of similar individuals, common opinions, and changes in social interactions based on a globally impactful event, using community detection algorithms on data extracted from comments and posts on the social network Reddit about the Russia-Ukraine war between 2022 and 2023. Initially, the data undergoes a characterization process to identify its relevant properties, which then aids in the final objective of this work - modeling graphs that will allow for deeper observation and analysis of the communities present in the investigated material.

Keywords: Community Detection. Opinion Networks. Reddit. Russo-Ukrainian War. Data Characterization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LIWC	Linguistic Inquiry and Word Count
VADER	Valence Aware Dictionary and sEntiment Reasoner

SUMÁRIO

1 INTRODUÇÃO

Com acessibilidade e capacidade de disseminar informação rapidamente, as redes sociais se tornaram uma ferramenta fundamental de observação da cultura humana, das opiniões e posicionamentos de diferentes círculos sociais, e da repercussão gerada por qualquer evento ao redor do mundo. Qualquer usuário, independentemente de status, fama ou influência, pode expor seu ponto de vista sobre determinado tópico para que outros usuários vejam e interajam, por sua vez adicionando as próprias impressões à discussão. Assim como no mundo real, isso promove a criação de relações interpessoais, convergindo e divergindo indivíduos com base em suas interações, para eventualmente levar a criação de círculos sociais dentro do próprio ambiente virtual Elena-Iulia (2018).

Naturalmente, o compartilhamento de interesses e opiniões similares faz com que usuários convirjam em grupos, que podemos denominar de comunidades. Segundo (WELLMAN, 1997), comunidades são redes de laços interpessoais que proporcionam sociabilidade e identidade social. Dessa forma, as comunidades no meio virtual são agregações de opiniões que, considerando a facilidade de acesso da mídia, podem ser analisadas para extrair informações sobre qualquer assunto desejado.

Com o surgimento destes agrupamentos em mente, alguns autores propõem a utilização de algoritmos de detecção de comunidades em redes sociais que são baseados em teoria dos grafos na ciência da computação e em agrupamentos hierárquicos na sociologia (NEWMAN; GIRVAN, 2004). O objetivo é utilizar técnicas para encontrar e modelar comunidades dentro de um conjunto de dados em formato de grafo, onde os nodos correspondem aos usuários, e as arestas às interações entre eles. Nesse contexto, comunidades são vértices que compartilham propriedades comuns, podendo assim um mesmo vértice fazer parte de múltiplas comunidades. É por meio da visualização destes grafos que se torna possível, por exemplo, auxiliar as mídias sociais a direcionarem algum conteúdo específico para usuários que têm maior probabilidade de se interessarem por ele, ou também tentar unir usuários que compartilham de interesses similares.

Um exemplo da união de indivíduos e direcionamento de conteúdo baseado em seu interesses acontece na rede social Reddit[1], que tenta direcionar seus usuários para fóruns mais interessantes para cada pessoa de acordo com os interesses demonstrados por ela. Consequentemente, usuários que consomem conteúdo similar tendem a frequentar páginas similares dentro da rede.

O Reddit é uma rede social dividida em "subreddits", comunidades criadas por usuários normalmente dedicadas ao compartilhamento e discussão de conteúdo relacionado a algum tema específico. Segundo (HORNE; ADALI; SIKDAR, 2017, p. 1):

"Reddit é uma das mais populares plataformas de compartilhamento e

[1] https://www.reddit.com/

> *discussão, classificada como #4 nos Estados Unidos e #16 no resto do mundo. Reddit alega ser a front page da internet".*

Devido à sua popularidade e também ao fato de todo o material compartilhado vir dos próprios usuários, essa rede se tornou um excelente meio de adquirir opiniões e informações referentes à guerra, tema de impacto global que consequentemente provocou grande repercussão no meio virtual. Espera-se com este trabalho compreender não apenas os tópicos e opiniões que surgiram decorrente da guerra, como também as variáveis que influenciam as discussões dentro da rede social sobre esse tema. Utilizando conceitos de computação social, será possível identificar as condições que influenciam a disseminação de algum determinado tipo de conteúdo dentro de uma comunidade virtual. Serão aplicados algoritmos de detecção de comunidade sobre os dados coletados, para que por fim as redes de opinião presentes possam ser detectadas e analisadas.

1.1 OBJETIVO GERAL

Este projeto tem como meta utilizar os dados coletados do Reddit sobre a guerra da Ucrânia entre os períodos de Fevereiro de 2022 a Julho de 2023 para realizar detecção de comunidades por meio da construção de grafos. Consequentemente, estes grafos servirão de auxílio para compreender melhor as opiniões presentes dentro do conjunto inicial de dados.

1.2 OBJETIVOS ESPECÍFICOS

Considerando o objetivo geral previamente descrito, consideram-se objetivos específicos:

- Extração de dados do Reddit utilizando técnicas de *web scraping*;
- Caracterização dos dados extraídos, utilizando principalmente a geração de gráficos como mapas de calor, nuvens de palavras, análises quantitativas e análises de sentimento;
- Modelagem de grafos baseados nas árvores de comentários e respostas de postagens dentro dos dados coletados;
- Aplicação de algoritmos de detecção de comunidades sobre os grafos;

1.3 METODOLOGIA

A metodologia é baseada nas seguintes etapas:

- **Estudo e análise dos conceitos básicos e dos trabalhos correlatos:**
  Estudar os conceitos necessários para a progressão do trabalho. Identificar as ferramentas necessárias para o desenvolvimento e organizar e planejar a estrutura geral do material produzido. Além disso, estudar trabalhos similares para entender o estado-da-arte e obter a fundamentação necessária para realizar o desenvolvimento. Os passos desta etapa são:

  – Definir quais ferramentas serão necessárias para o desenvolvimento;
  – Mapear a estrutura geral do trabalho;
  – Pesquisar e ler trabalhos similares.

- **Coleta de dados da rede social:**
  Desenvolver um software para coletar dados de comentários e postagens da rede social Reddit, referentes ao tema da Guerra entre Rússia e Ucrânia, utilizando técnicas de *web scraping*, ou seja, extração automatizada de conteúdo disponível na web. Consistem em atividades desta etapa:

  – Definir as páginas web de onde serão extraídos os dados;
  – Desenvolver um software que extrai os dados e os armazena em formato JSON e CSV.

- **Caracterização e análise inicial dos dados extraídos:**
  Aplicar diversas técnicas de caracterização dos dados coletados, a fim de compreender melhor o conteúdo presente neles. Para isso, desenvolve-se *scripts* que geram gráficos úteis no processo de análise, incluindo nuvens de palavras, mapas de calor e redes de co-ocorrência. A partir da visualização dos gráficos gerados, e da análise textual do conteúdo presente nos dados coletados, é visada a compreensão aprofundada do material que foi extraído, no contexto dos usuários da rede social e do tema em questão. São atividades desta etapa:

  – Desenvolver *scripts* para gerar gráficos, a partir dos dados armazenados;
  – Analisar o contexto da Guerra da Ucrânia;
  – Analisar o contexto das postagens e comentários feitos dentro da rede social;
  – Detalhar a comparação e observação do material textual extraído e os gráficos gerados a partir dele.

- **Modelagem em grafos e detecção de comunidades:**
  Modelar o conjunto de dados em formato de grafo, onde os vértices representam usuários e as arestas representam interações realizadas entre estes usuários. A partir disso, aplicar algoritmos de detecção de comunidade nas redes geradas, de forma

que seja possível visualizar os conjuntos de nodos que partilham de características semelhantes, e realizar uma análise mais aprofundada dos comportamentos e opiniões em destaque dentro de cada comunidade encontrada. As atividades desta etapa são:

- Analisar a utilização de diferentes técnicas de detecção de comunidades dentro de um conjunto de dados;
- Desenvolver *scripts* para gerar grafos a partir do conjunto de dados coletados;
- Realizar a detecção de comunidades nas redes modeladas;
- Realizar a análise das comunidades detectadas.

1.4 ESTRUTURA DO TRABALHO

O Capítulo 1 consiste em introdução, juntamente com objetivos gerais e específicos. O Capítulo 2 contém a fundamentação teórica que é necessária para a compreensão total dos conceitos expostos. O Capítulo 3 descreve outros trabalhos acadêmicos cujos resultados foram essenciais para a elaboração desta monografia. O Capítulo 4 contém a descrição da aplicação de diferentes métodos de caracterização sobre os dados coletados. O Capítulo 5 expõe análises realizados sobre grafos gerados a partir dos dados coletados, relata o processo de detecção de comunidades nesses grafos e apresenta análises sobre os resultados obtidos. Por fim, no Capítulo 6 são apresentadas as considerações finais e possíveis trabalhos futuros relacionados.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos necessários para a compreensão deste trabalho.

2.1 REDES SOCIAIS

Uma rede social nada mais é do que uma plataforma virtual onde usuários podem interagir com outros usuários. Segundo (ZENHA, 2018), a base do ambiente virtual é a interação síncrona e assíncrona de indivíduos, que exercem papel protagonista nas relações que estabelecem na rede. A autora também complementa que as redes constituem um conjunto de nós interconectados e dinâmicos, onde cada nó é um indivíduo e sua força é determinada pela quantidade de conexões que estabelece.

As redes sociais são propícias para a interação com pessoas conhecidas e desconhecidas. Além disso, se tornaram um ambiente excelente para o monitoramento de eventos impactantes ao redor do globo, devido a transmissão de informação sem limites de tempo e distância.

2.1.1 Usuário

Em uma rede social, um usuário é um indivíduo que acessa a plataforma para interagir com outros indivíduos, compartilhando informação textual ou audiovisual. Tipicamente, cada rede social promove seus usuários a interagir de alguma forma determinada: postagens de suas vidas pessoais, guias e tutoriais sobre os mais variados assuntos, comércio e publicidade, dentre outros.

Cada usuário determina que tipo de conteúdo quer expor em sua página pessoal para que outros usuários possam visualizar e interagir com ele, criando assim um perfil próprio, que é como cada indivíduo decide apresentar sua identidade dentro de um determinado intervalo de tempo e considerando a sua audiência (BHUTKAR, 2009).

2.1.2 Reddit

O Reddit é uma rede social fundada em 2005, com popularidade ascendente desde então. Seus usuários são majoritariamente falantes da língua inglesa, que é o idioma considerado oficial dentro da plataforma, mas seu alcance e influência são globais. Em meados de 2015, o Reddit possuía mais de 150 milhões de usuários únicos em mais de 200 países, com cerca de 1 milhão de acessos diários (ANDERSON, 2015).

A característica determinante do Reddit é o fato de ele ser organizado em subreddits, que são comunidades criadas e moderadas pelos próprios usuários, sempre focadas em um determinado tema, seja ele um filme, um país, uma celebridade ou qualquer assunto permitido pelas diretrizes da plataforma. Dentro de um subreddit, usuários podem fazer postagens sobre o tema relevante, e outros usuários podem interagir com as postagens

fazendo comentários ou "votando" nela, com um voto positivo (upvote) ou negativo (downvote), criando assim *threads* (tópicos) de discussão sobre o conteúdo compartilhado. Em geral, quanto mais upvotes uma postagem recebe, mais facilmente ela será mostrada para outros usuários que acessam a rede.

É importante ressaltar que o Reddit é uma rede majoritariamente anônima. Usuários possuem contas descartáveis usadas para expressar pensamentos e informações que por vezes podem ser tópicos sensíveis para a sociedade em geral (CHOUDHURY; DE, 2014).

2.2 CARACTERIZAÇÃO DE DADOS

A caracterização é um processo dentro da análise de dados que consiste em aplicar algumas técnicas que ajudam a identificar padrões e tendências dentro de um conjunto de dados. A caracterização é extremamente útil para auxiliar as tomadas de decisão quando se trabalha com uma base de dados, pois por meio dela é possível compreender melhor o conteúdo presente nesses dados.

O processo de caracterização pode fornecer informações descritivas, quantitativas e semânticas sobre o conjunto de dados relevantes. Para essa finalidade, são utilizados valores estatísticos que permitem visualizar a distribuição dos dados, gráficos para facilitar a identificação de tendências, e também algoritmos avançados para a detecção de padrões e de grupos de dados com características semelhantes.

Um conjunto de dados extraídos de uma rede social representa um grande emaranhado de opiniões e pontos de vista diferentes, dispostos de forma caótica em meio a tantas discussões sobre uma infinidade de tópicos. Em decorrência disso, a caracterização é um processo essencial para determinar com mais precisão característica importantes destes dados, como sua relevância, sua repercussão, e seu significado semântico, ou seja, qual mensagem ele transmite dentro do contexto em que está inserido. Dessa forma, é essencial submeter os dados coletados a este processo de caracterização, de modo que as fases subsequentes do trabalho possam ser desenvolvidas com consideração sobre o material base.

2.2.1 Técnicas de caracterização de dados

Nesta seção são descritos métodos que foram utilizados neste trabalho para caracterizar os dados extraídos.

- **Nuvem de palavras:**
  Consiste em uma forma visual de representar as palavras dentro de um texto, de acordo com a frequência de cada palavra. Quanto maior o tamanho da palavra em uma nuvem de palavras, mais vezes ela foi encontrada dentro do conjunto de

dados analisados. Para realizar a geração das nuvens de palavras apresentadas neste projeto, a biblioteca WordCloud[1] do Python foi utilizada.

- **VADER:**[2]
  É uma biblioteca do Python responsável por fazer análise de sentimentos de palavras, especialmente de sentimentos expressados comumente em redes sociais. Dado um texto de entrada, a ferramenta retorna os valores pos, neg, neu, que respectivamente correspondem a números representado a porcentagem de valores positivos, negativos e neutros encontrados no texto. Por fim, também retorna um valor compound, que representa uma medida unidimensional de sentimento de uma sentença, variando de -1 a 1, onde valores maiores que 0.05 tem sentimento positivo e menores que -0.05 tem sentimento negativo (HUTTO; GILBERT, 2014).

- **Análise LIWC:**[3]
  É uma ferramenta para análise de texto que pega palavras e as coloca em categorias gramaticais. Segundo (CARVALHO *et al.*, 2019), é possível obter variados tipos de informação de usuários de redes sociais utilizando LIWC, como tendências políticas e status sócio-econômico. Os autores complementam que o LIWC faz a análise componentes estruturais, sociais e cognitivos do texto e associa-os a categorias relacionadas. Dessa forma, essa ferramenta se mostra um grande auxílio para compreender o contexto semântico dos dados sendo analisados.

- **Teste de Kruskal-Wallis:**
  Consiste em um método para analisar pelo menos três grupos independentes de dados não-conformes com a distribuição normal. Esse teste compara as classificações dos dados entre os grupos determinados.

  O teste é feito em etapas. A primeira envolve assumir uma hipótese nula onde não há variação significativa entre os grupos sobre uma variável escolhida, e uma variável alternativa, onde pelo menos um grupo difere significativamente. Após isso, os dados são classificados e ranqueados, para posteriormente serem comparados com um valor crítico baseado na distribuição normal.

  Quando a estatística calculada pelo teste é maior que o valor crítico, a hipótese nula é rejeitada, ou seja, conclui-se que existe pelo menos uma diferença significativa entre os grupos. Caso a estatística seja menor que o valor crítico, não existem evidências suficientes para concluir que os grupos diferem de forma significativa.

- **Coeficiente de Gini:**
  Normalmente, o coeficiente de Gini é uma medida de desigualdade econômica, usada

[1] `https://pypi.org/project/wordcloud/`
[2] `https://pypi.org/project/vaderSentiment/`
[3] `https://pypi.org/project/liwc/`

para quantificar a disparidade de renda entre indivíduos dentro de uma designada população. A medida varia entre 0, que representa uma perfeita igualdade entre os indivíduos avaliados, e 1, que representa disparidade total (no caso de uma avaliação de renda, um indivíduo possui absolutamente toda a renda).

Também é possível utilizar o coeficiente de Gini para medir a desigualdade de valores em contextos diferentes. Dentro deste trabalho, o coeficiente é usado como complemento da análise LIWC, para identificar quais foram os atributos mais discrepantes encontrados.

- **Rede de Co-ocorrência:**
  Consiste em um grafo usado para demonstrar visualmente a ocorrência conjunta de elementos dentro de um agrupamento de dados. Os nodos do grafo representam cada elemento sendo analisado, enquanto que as arestas representam as ocorrências conjuntas dos nodos sendo conectados por elas.

  As redes são uma ferramenta muito útil na caracterização dos dados, bem como em algoritmos de mídias sociais, para auxiliar na identificação e na correlação de grupos que possuem perfis similares. No contexto deste trabalho, é útil para relacionar palavras que aparecem juntas frequentemente dentro dos dados que foram extraídos de postagens e comentários.

2.3 DETECÇÃO DE COMUNIDADES

2.3.1 Grafos

Na matemática, grafos são estruturas comumente utilizadas como forma de visualizar conjuntos de objetos, representados por nodos, e as relações existentes entre estes objetos, representadas por arestas. Estes componentes presentes em grafos podem fornecer estatísticas úteis sobre o conjunto sendo visualizado, como o grau de cada nodo (quantidade de arestas ligadas a cada nodo) e o peso das arestas (normalmente relacionado a quantidade de vezes que a conexão representada por cada aresta se repete). Além disso, grafos podem ser direcionados, situação em que as arestas possuem uma "direção" determinada (indo de nodo A para nodo B).

No contexto deste trabalho, os grafos são empregados para auxiliar a análise da rede social Reddit e a detecção de comunidades. Nesse sentido, os nodos representam usuários da rede social, e as arestas representam um comentário direcionado de um usuário para o outro. No caso de um usuário realizar múltiplos comentários para um mesmo indivíduo, o peso das arestas aumenta com cada comentário.

2.3.2 Comunidades

Para conceituar comunidades, é importante visualizá-las dentro do contexto de grafos. Uma estrutura de comunidade consiste em um subconjunto de nodos dentro de um grafo que são densamente interconectados entre si, enquanto possuem uma quantidade menor de conexões com outros nodos fora deste subconjunto.

Trazendo isso para o contexto de redes sociais, uma comunidade pode representar um grupo de pessoas que possuem interesses similares, uma rede de amigos, um grupo de pessoas que compartilha o mesmo local de convivência, ou laços de família, por exemplo. (HE; CHEN, 2015).

2.3.3 Modularidade

A modularidade é uma medida usada para qualificar uma divisão de rede de grafos em comunidades. Ao separarmos a rede em comunidades menores, ou seja, em sub-grafos dentro de um grafo maior, é possível utilizar a modularidade para quantificar o grau de proximidade das conexões dentro de uma comunidade com o grau esperado caso essas conexões estivessem distribuídas de forma aleatória. Assim, numa escala de -1/2 a 1, quanto maior for o valor da modularidade, melhor é a divisão em comunidades realizada, ou seja, maior é a probabilidade de que as comunidades de fato possuam características similares entre nodos que estão conectados. A fórmula da modularidade é tipicamente dada por:

$$Q = \frac{1}{2m} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(A_{ij} - \frac{kikj}{2m} \right) \delta(c_i, c_j)$$

Na equação acima, "Q" é o caractere que representa o resultado de modularidade. "m" é a soma de todos os pesos das arestas do grafo, N é o número total de nodos no grafo, "A_{ij}" é o peso da arestas conectando os nodos i e j, "k_i" e "k_j" são a soma dos pesos das arestas conectadas a i e j, respectivamente, e $\delta(c_i, c_j)$ é uma função delta que retorna 1 se a comunidade de i (c_i) e a comunidade de j (c_j) forem a mesma, e 0 caso contrário.

2.3.4 Algoritmo de Louvain

O algoritmo de Louvain é um método de identificar comunidades em grandes estruturas de rede em grafos. O algoritmo consiste em duas fases. A primeira, chamada de otimização de Modularidade, insere cada nodo de um grafo em subgrafos (comunidades) diferentes, até que não ocorra um aumento significativo de modularidade. Neste trabalho, a implementação do algoritmo vem da biblioteca CDLib em Python.[4]. A segunda etapa percorre as comunidades identificadas na primeira. Os nodos de cada comunidade são

[4] `https://cdlib.readthedocs.io/en/latest/`

todos agrupados em um só, e dessa forma a primeira etapa se repete com os novos nodos, até que um máximo global de valor de modularidade seja alcançado.

O algoritmo visa ter um tempo computacional eficiente relativo a outros algoritmos de detecção de comunidades, além de proporcionar acurácia em relação às comunidades detectadas, devido ao uso do valor de modularidade como parâmetro para julgar a qualidade das comunidades detectadas (BLONDEL *et al.*, 2008).

2.3.5 Algoritmo de Leiden

O algoritmo de Leiden é, em sua essência, uma adaptação do algoritmo de Louvain. Inicialmente, cada nodo da rede recebe uma comunidade separada. O algoritmo itera então sobre cada nodo, movendo-os para comunidades vizinhas e observando se houve um aumento de modularidade, até que mais nenhum movimento provoque este aumento. Novamente, a implementação deste algoritmo no trabalho é realizada pela biblioteca CDLib. Depois desta etapa inicial, os nós de cada comunidade serão agrupados e, diferentemente do algoritmo de Louvain, acontecerá uma etapa de refinamento das comunidades, que serão avaliadas novamente para determinar se algum movimento gera um aumento de modularidade. Essas etapas de otimização e refinamento são repetidas pelo algoritmo de Leiden até que ele determine que não ocorrerá mais aumento na modularidade.

Segundo (TRAAG; WALTMAN; ECK, 2019), o algoritmo de Leiden é mais rápido que Louvain computacionalmente, além de garantir que as comunidades encontradas estão de fato conectadas. Em suma, este algoritmo pode ser mais rápido e garantir que partições melhores sejam identificadas.

3 TRABALHOS RELACIONADOS

Este trabalho utilizou alguns conceitos de análise de redes sociais, análise de sentimentos e detecção de comunidades que estão fortemente relacionados com outros trabalhos similares. Esta seção apresentará alguns destes trabalhos, para esclarecer as atividades já desenvolvidas nessa área de pesquisa e pontuar os princípios que influenciaram direta ou indiretamente este trabalho.

3.1 REDDIT COMO FERRAMENTA DE ANÁLISE

Alguns trabalhos utilizam o Reddit como ferramenta de obtenção e análise de dados, devido à sua popularidade e diversidade de usuários e *subreddits*. Para que isso seja possível, é fundamental sumarizar aspectos relevantes do *site*, como no trabalho de Anderson (2015), que exemplifica o funcionamento do site, juntamente com o perfil dos usuários e as vantagens e desvantagens de sua utilização como ferramenta de compartilhamento de informação em escala global, fornecendo um sumário útil do *site*.

Um dos aspectos fundamentais para o desenvolvimento deste trabalho é o uso do Reddit para coleta e extração de dados. Nesse sentido, torna-se importante considerar alguns aspectos como ferramentas de extração mais apropriadas, quais atributos serão mais relevantes para as análises, quais tópicos de estudo são mais interessantes e, naturalmente, os aspectos éticos do uso dessa rede como material de pesquisa. Todos esses pontos são objeto de estudo no trabalho de Proferes *et al.* (2021), que destaca algumas características que devem ser consideradas por pesquisadores ao trabalhar com essa rede, como a cultura muito variada nos diferentes fóruns do *site*.

Outro perspectiva importante a ser considerada é a pesquisa e análise na área de comunicações e interações sociais. A organização complexa dos círculos sociais do Reddit e o fato de que praticamente todo o conteúdo compartilhado vem dos próprios usuários naturalmente fornece um ambiente propício para a coleta e o estudo de dados. Entretanto, para que o estudo seja possível em primeiro lugar, é de suma importância que o pesquisador entenda a cultura não apenas do *site* como dos *subreddits* individuais dentro dele, como é mencionado no trabalho de Hintz e Betts (2022).

Finalmente, fora do escopo particular do Reddit, também existe a preocupação com os preceitos éticos que devem ser considerados ao utilizar dados de usuários coletados de qualquer rede social. O ambiente virtual tem, fundamentalmente, um grau de anonimidade em relação ao conteúdo compartilhado. Além disso, a interpretação dos pontos de vista expostos em uma discussão *online* é mais complicada, por envolver textos e imagens no lugar de palavras e linguagem corporal. É pensando nisso que o trabalho de Gliniecka (2023) sumariza diversos questionamentos importantes para pesquisadores de redes sociais, visando facilitar a visualização dos diversos aspectos que devem ser considerados quando se trabalho com dados virtuais vindos de plataformas de comunicação livre.

3.2 A GUERRA RUSSO-UCRANIANA NAS REDES SOCIAIS

Como mencionado anteriormente neste trabalho, as redes sociais são úteis para monitorar a reação dos usuários a eventos de grande impacto mundial. Naturalmente, a guerra entre Rússia e Ucrânia não foi uma exceção, provocando a criação de diversas comunidades dentro do Reddit dedicadas à discussão e compartilhamento de informação sobre o conflito. Além disso, também surgiram diversos trabalhos acadêmicos inspirados no tema.

As redes sociais, devido ao seu alcance e prevalência global, também servem como ferramenta para os próprios países participantes da guerra. Esse aspecto é bastante explorado no trabalho de Ciuriak (2022), exemplificando algumas ações tomadas pela Rússia, por exemplo, para criar uma névoa virtual e gerar incerteza sobre suas ações durante o conflito. A percepção do público em geral é grandemente influenciada pelo conteúdo compartilhado nas redes, que naturalmente se torna uma preocupação bem grande para países em estado de guerra.

Para um estudo apropriado do impacto nas guerras nas redes sociais, um passo inicial importante logicamente é identificar os locais virtuais de onde é possível extrair informação relevante. Este foi um ponto considerado durante o desenvolvimento deste trabalho, mas grandemente auxiliado por trabalhos como o de Zhu *et al.* (2022), que sumariza quais comunidades dentro do Reddit são mais relevantes para a coleta de dados referentes à guerra.

A análise de sentimentos de dados coletados em redes sociais em geral é bastante documentada por uma diversidade gigantesca de trabalhos acadêmicos. Entretanto, realizar essa análise direcionada para o Reddit e tratando do tópico da Guerra Russo-ucraniana, que ainda persiste mesmo durante o desenvolvimento deste trabalho, provê desafios e pontos mais específicos a serem considerados. O trabalho de Guerra e Karakuş (2023) é um exemplo de trabalho recente sobre detecção de sentimentos em comunidades do Reddit sobre a guerra entre Rússia e Ucrânia, e a metodologia descrita é um bom exemplo para trabalhos com temas similares.

3.3 ANÁLISE E DETECÇÃO DE COMUNIDADES POR MEIO DE GRAFOS

A seção de desenvolvimento deste trabalho utiliza grafos para realizar a detecção de comunidades sobre os dados coletados do Reddit. Este processo é, na verdade, um uso comum da estrutura de grafos, que aparece em diversos outros trabalhos acadêmicos para atingir objetivos similares ao deste. Adicionalmente, compreender a teoria dos grafos, explicada em trabalhos como o de Fortunato (2010), é de suma importância para que a análise de sua estrutura seja possível.

A utilização de algoritmos de detecção de comunidades em redes sociais também necessita de um estudo mais aprofundado, devido à imensa diversidade de técnicas e

algoritmos diferentes, que geram resultados variados em termos de eficiência e conteúdo gerado. Trabalhos como o de Chunaev (2020) e Bedi e Sharma (2016) sumarizam diversos algoritmos, permitindo uma visão mais clara de qual deles são apropriados ou não para alcançar os resultados desejados em um trabalho acadêmico que explora a detecção e análise de comunidades.

3.4 ANÁLISE COMPARATIVA

A fim de sumarizar e facilitar a visualização do material apresentado em cada um dos trabalhos citados e quais foram suas contribuições para este trabalho, foi criada a tabela 1. Ao observá-la, é perceptível a quantidade de material já existente sobre o Reddit, a guerra Russo-ucraniana e o processo de detecção de comunidades a partir de dados extraídos de redes sociais. A coluna “Contribuições” visa pontuar de forma mais específica quais aspectos de cada trabalho citado serviram de base para o desenvolvimento deste.

Autores	**Ano**	**Título do Trabalho**	**Rede social tema**	**Método de pesquisa**	**Contribuições**
Katie	2015	Ask me anything: what is Reddit?	Reddit	Pesquisa bibliográfica	Contextualizar o funcionamento e estrutura do Reddit
Proferes, Jones, Gilbert, Fiesler e Zimmer	2021	Studying Reddit: A Systematic Overview of Disciplines, Approaches, Methods, and Ethics	Reddit	Pesquisa bibliográfica	Determinar ferramentas e atributos relevantes para a coleta e estudos de dados do Reddit
Hintz e Betts	2022	Reddit in communication research: current status, future directions and best practices	Reddit	Pesquisa bibliográfica	Determinar abordagens apropriadas para o estudo do Reddit, levando em conta suas subculturas virtuais
Gliniecka	2023	The Ethics of Publicly Available Data Research: A Situated Ethics Framework for Reddit	Reddit	Pesquisa bibliográfica	Elaborar uma estrutura referencial para uso de dados públicos extraídos do Reddit
Ciuriak	2022	The Role of Social Media in Russia's War on Ukraine	Twitter, Reddit, Instagram e redes sociais em geral	Pesquisa bibliográfica	Determinar a influência do conflito entre Rússia e Ucrânia nas interações em redes sociais
Zhu, Haq, Lee, Tyson e Hui	2022	A Reddit Dataset for the Russo-Ukrainian Conflict in 2022	Reddit	Pesquisa bibliográfica e exploração de fóruns dentro do Reddit	Sumarizar as fontes mais úteis de conteúdo relevante à guerra dentro do Reddit
Guerra e Karakuş	2023	Sentiment analysis for measuring hope and fear from Reddit posts during the 2022 Russo-Ukrainian conflict	Reddit	Extração e análise de sentimentos de dados do Reddit sobre a guerra entre Rússia e Ucrânia	Identificar técnicas úteis e práticas apropriadas de detecção de sentimentos no Reddit.
Fortunato	2010	Community detection in graphs	Não se aplica	Estudo de teoria de grafos e detecção de comunidades	Compreender o funcionamento das estruturas de grafos e técnicas de detecção de comunidades
Chunaev	2020	Community detection in node-attributed social networks: A survey	Redes sociais em geral	Enquete online	Determinar as técnicas mais apropriadas para serem utilizadas na detecção de comunidades em grafos
Bedi e Sharma	2016	Community detection in social networks	Redes sociais em geral	Estudo da teoria de algoritmos de detecção de comunidades	Entender o funcionamento de algoritmos de detecção de comunidades

Tabela 1 – Sumário dos trabalhos relacionados.

4 CARACTERIZAÇÃO DE DADOS

Este capítulo do trabalho apresenta as análises feitas sobre o conjunto de dados extraídos, aplicando os métodos de caracterização que foram explicados na seção 2.2.1.

4.1 METODOLOGIA

Primeiramente, os dados receberam uma análise quantitativa em relação às postagens e comentários feitos dentro delas. Posteriormente, também é feita a correlação de termos relevantes dentro do texto do conteúdo que foi extraído, por meio de nuvens de palavra e rede de co-ocorrência. Também são utilizadas ferramentas como VADER e LIWC para identificar características semânticas, como os sentimentos expressados pelos usuários no texto e a análise psico-linguística, onde ambos o teste de Kruskal-Wallis e o coeficiente de Gini foram utilizados para identificar os resultados de maior relevância para o trabalho.

4.2 COLETA DE DADOS

O processo de coleta de dados foi realizado através de *web scraping*, ou seja, extração de conteúdo diretamente de páginas web. Visando esse objetivo, foram criados *scripts* de coleta usando majoritariamente duas ferramentas: a API própria do Reddit[1], disponível para desenvolvedores em geral e que permite extrair conteúdo de praticamente qualquer lugar no site[2], juntamente com o framework Selenium[3], utilizado para automatizar a procura por conteúdo sobre a guerra entre Rússia e Ucrânia.

O conjunto de dados extraído para as análises apresentadas neste trabalho consiste de postagens e comentários coletados de comunidades (*subreddits*) do Reddit referentes ao período da semana de 20 de Fevereiro de 2022, início da invasão russa na Ucrânia, até a semana de 20 de Junho de 2023, data aproximada do início deste projeto de TCC. Foram escolhidos subreddits que continham conteúdos mais relevantes ou direcionados ao contexto da guerra.

Para o caso de comunidades que tinham muitas postagens não relacionadas à guerra, o conteúdo extraído foi filtrado para selecionar apenas material contendo palavras-chave, como *Biden, Trump, Putin, Zelensky, Ukraine, Ukrainian, Russia, Russian, Ukraine War, Ukraine-Russia War, Kyiv, Crimea, Snake Island* e *Moscow*. Essa filtragem foi feita tanto utilizando essas palavras-chave como critério na busca automatizada de dados do Reddit por Selenium, quanto percorrendo os dados que já tinham sido extraídos e filtrando aqueles sem conteúdo relacionado às palavras-chave.

[1] Disponível em: `https://www.reddit.com/dev/api/`

[2] A API do Reddit sofreu algumas alterações na regulamentação após o período de extração de dados deste trabalho, impondo uma taxa de utilização que antes não existia.

[3] Disponível em: `https://www.selenium.dev/`

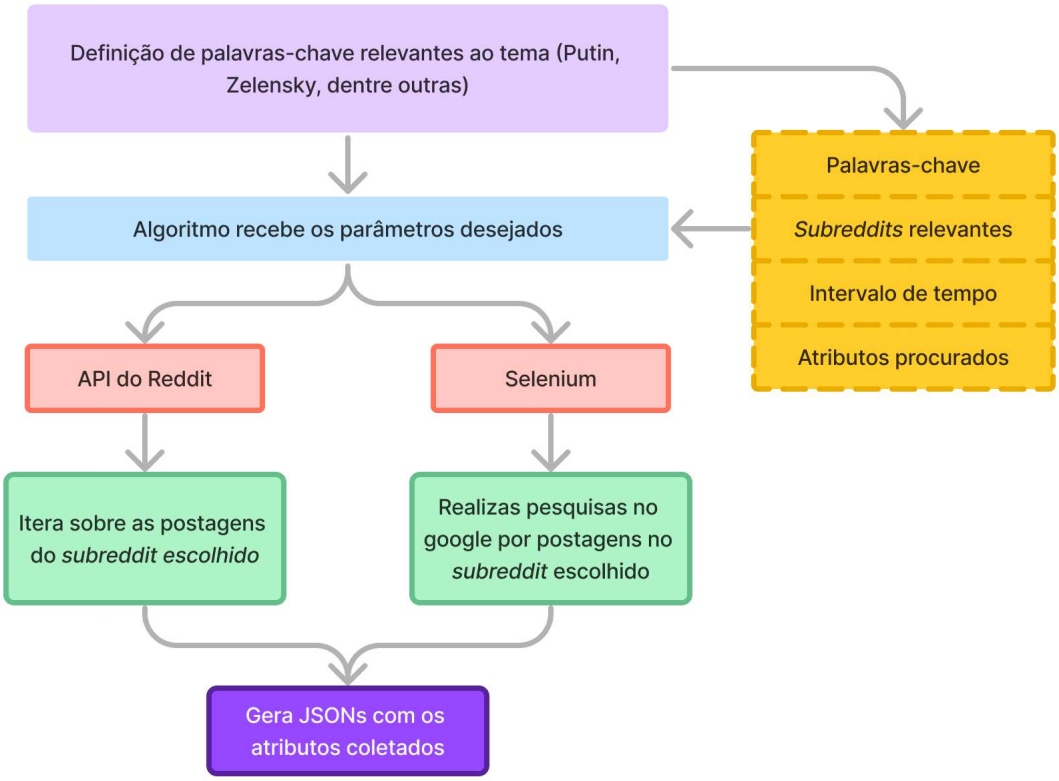
Abaixo estão listadas as comunidades das quais os dados foram coletados:

- r/EndlessWar: comunidade dedicada a discussão sobre guerras em geral. Durante o período de coleta deste trabalho, o conteúdo postado era predominantemente sobre o conflito Russo-Ucraniano;
- r/News: comunidade dedicada ao compartilhamento de notícias mundiais, mas com certo foco em material relevante aos Estados Unidos;
- r/WorldNews: comunidade dedicada ao compartilhamento de notícias mundiais, com conteúdo menos direcionado que a comunidade citada acima;
- r/Politics: comunidade dedicada à discussão de eventos relevantes sobre a política mundial;
- r/RussiaUkraineWar2022: comunidade criada após a invasão russa, dedicada à discussão e compartilhamento de conteúdo relevante sobre a guerra;
- r/Ukraine: inicialmente uma comunidade primariamente voltada a ucranianos, após o início da guerra se tornou um fórum global dedicado à discussão e transmissão de informação relevante ao conflito, bem como um meio de suporte à Ucrânia em geral;
- r/UkraineWarVideoReport: comunidade criada após a invasão, também dedicada ao compartilhamento de conteúdo relevante.

Os scripts foram utilizados sobre todas as comunidades listadas acima, coletando postagens e comentários de acordo com critérios de relevância (quantidade de votos recebidos dentro do Reddit) e tempo. O material coletado das postagens foi guardado em formato JSON, organizado dentro das seguintes chaves: Título da postagem, Autor (usuário que postou), ID (código único de cada postagem usado pelo Reddit), URL, Data, Pontuação (votos recebidos dentro da rede), Conteúdo da postagem, Número de comentários e lista de comentários. A lista de comentários contém toda a árvore de respostas da postagem, incluindo respostas de respostas e assim adiante. Posteriormente, essas chaves também foram convertidas para o formato CSV para facilitar o uso dos dados.

A figura 1 apresenta de forma simplificada como a extração foi realizada, utilizando tanto a API do Reddit quanto o Selenium para gerar arquivos JSON com os atributos desejados.

Figura 1 – Sequência de passos realizados na extração de dados.



Fonte: Elaboração própria.

Abaixo, é apresentado um recorte de como os dados ficaram estruturados após a coleta, em formato JSON. Neste caso, observa-se uma postagem, com um comentário subsequente, e uma resposta para o comentário, demonstrando como funciona a árvore de comentários dentro de uma postagem.


```
{
        "Title": "Uncanny predictions of Ukraine's war from April 2021 by former Russian MP Nevzorov",
        "Author": "Ortenrosse",
        "ID": "taood7",
        "Url": "https://v.redd.it/ekwlm57vygm81",
        "Time": 1646880658.0,
        "Score": 9863,
        "Submission Content": "",
        "Number of Comments": 826,
        "Comment List": [
            {
                "Author": "Starter91",
                "ID": "i02rz97",
                "Comment Date": 1646893273.0,
                "Comment Score": 676,
                "Comment Content": "Is this man a time traveler?",
                "Number of Replies": 17,
                "Comment Replies": [
                    {
                        ...
```

4.3 ANÁLISE QUANTITATIVA DO CONJUNTO DE DADOS EXTRAÍDOS

Terminada a extração e a filtragem de conteúdo relevante, o conjunto de dados final consiste no texto referente a postagens e comentários das comunidades dentro do período mencionado. É importante ressaltar que, além do propósito diferente que cada comunidade possui dentro do Reddit, elas também têm um valor bem variado de número de usuários frequentadores, o que influencia nas análises quantitativas que serão apresentadas posteriormente. A Tabela 2 informa o tamanho, baseado na quantidade de usuários participantes de cada uma das comunidades selecionadas para o trabalho, logo após o período de coleta dos dados.

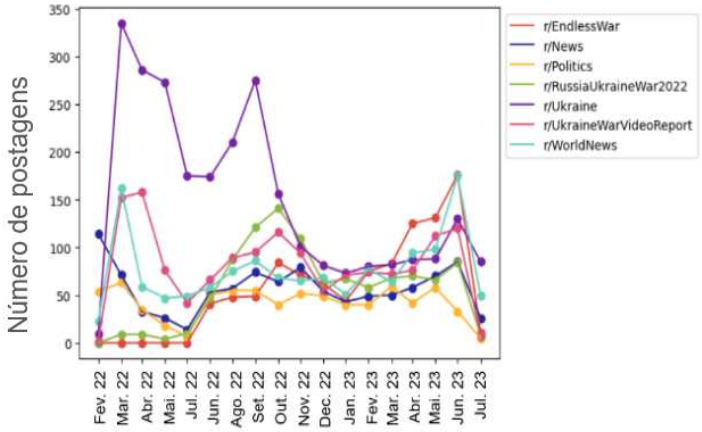
Tabela 2 – Número aproximado de usuários inscritos em cada *subreddit*.

Comunidade	Nº aprox. de usuários participantes
r/endlesswar	33.000
r/news	27.000.000
r/worldnews	34.000.000
r/politics	8.400.000
r/russiaukrainewar2022	211.000
r/ukraine	900.000
r/ukrainewarvideoreport	700.000

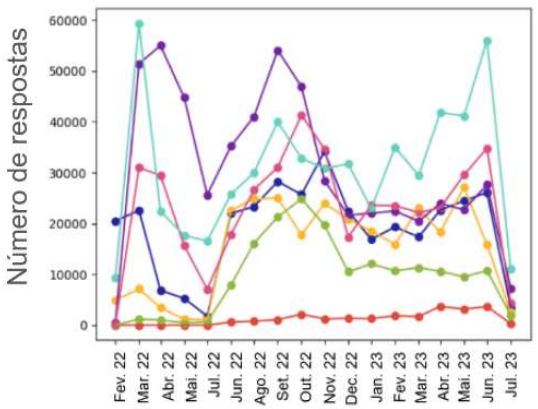
Fonte: Elaboração própria.

A Figura 2 representa a contagem do número total de postagens e comentários presentes em cada comunidade, em cada mês analisado. Quando a linha dos comentários está acima da de postagens em um mesmo período de tempo, pode-se interpretar que cada postagem teve um engajamento grande, dentro do contexto de cada comunidade. Esse fenômeno pode ser observado, por exemplo, na linha correspondente à comunidade *r/worldnews* durante o mês de Março, ponto inicial do conflito, onde é perceptível uma quantidade imensa de comentários quando comparado a quantidade de postagens.

Figura 2 – Contagem de postagens e respostas para cada mês de conflito.



(a) Contagem de postagens



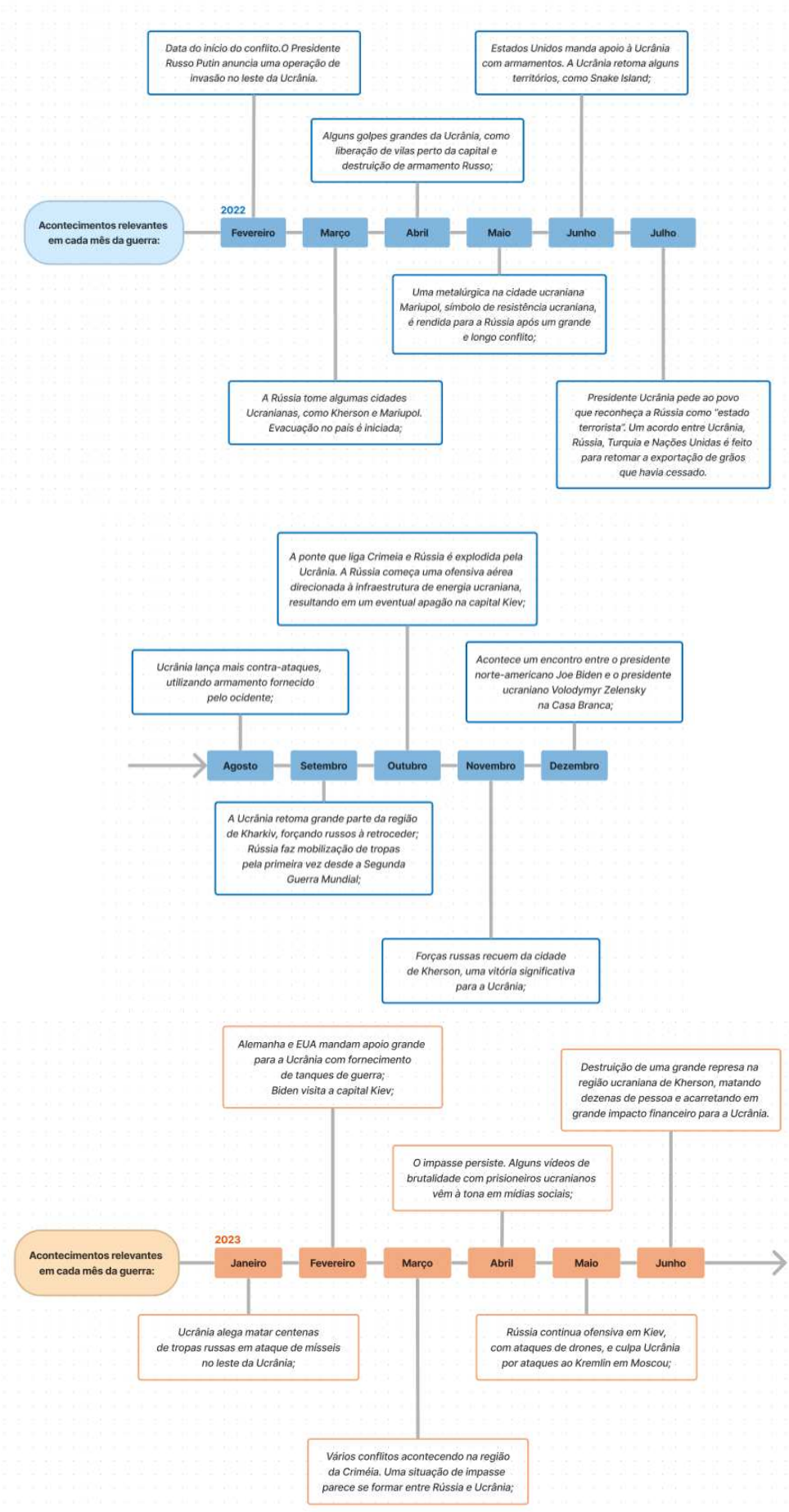
(b) Contagem de respostas

Fonte: Elaboração própria.

Por outro lado, se observarmos *r/endlesswar*, uma comunidade relativamente menor, é notável que a linha de comentários se mantém estável mesmo quando ocorre um pico na quantidade de postagens, indicando que houve poucas interações com cada postagem durante o período. Para o resto das comunidades, em geral, as linhas do gráfico sobem e descem de forma relativamente sincronizada.

Para analisar o comportamento dos dados em relação ao contexto da guerra, é ideal observar quais acontecimentos relevantes aconteceram em cada mês. A Figura 3 fornece, de maneira resumida, os acontecimentos que possivelmente estavam em alta nas discussões em mídias sociais para cada mês de conflito, como por exemplo as interações entre figuras políticas estadunidenses e ucranianas que ocorrem em determinados momentos do conflito, ou também a tomada de território por parte da Rússia.

Figura 3 – Linha do tempo de eventos relevantes do conflito.

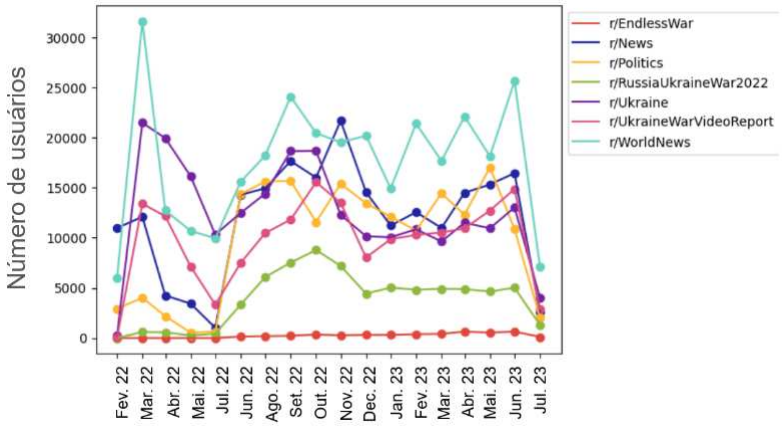


Fonte: Elaboração própria.

Naturalmente, durante o início do conflito tanto a quantidade de postagens quanto o volume de respostas em cada postagem são bem altos, principalmente nos *subreddits* dedicados à notícias. Logo vemos a ascensão de atividade em *subreddits* dedicados exclusivamente à guerra, e é perceptível que eles mantêm uma estatística relativamente estável durante todo o período analisado. Além do pico inicial, também é perceptível um pico por volta de Setembro e Outubro de 2022. Esse foi um período de grande contra-ataque ucraniano, e a comoção na rede social pode, dentre outros fatores, estar relacionado à predominância de público ocidental no Reddit, principalmente pessoas pertencentes aos Estados Unidos e União Europeia, apoiadores da Ucrânia durante a guerra. O último pico, observado próximo ao fim do intervalo de tempo analisado, pode ter sido influenciado tanto pelas grande perdas por parte da Ucrânia sofridas nestes meses.

Outra medida analisada foi a quantidade de usuários únicos por mês em cada comunidade, demonstrada na Figura 4. Um "usuário único", neste contexto, se refere a cada usuário que interagiu pelo menos uma vez no*subreddit* naquele mês. Nota-se que as comunidades menores também tem uma variação menor de usuários, possivelmente indicando usuários "fiéis", ou seja, que são frequentadores dessas comunidades em específico. Nas comunidades maiores, existe uma variação contínua, observada pela linha em zigue-zague, o que pode significar que usuários nem sempre interagem com conteúdo relacionado à guerra, mas participam da discussão quando eventos impactantes ocorrem.

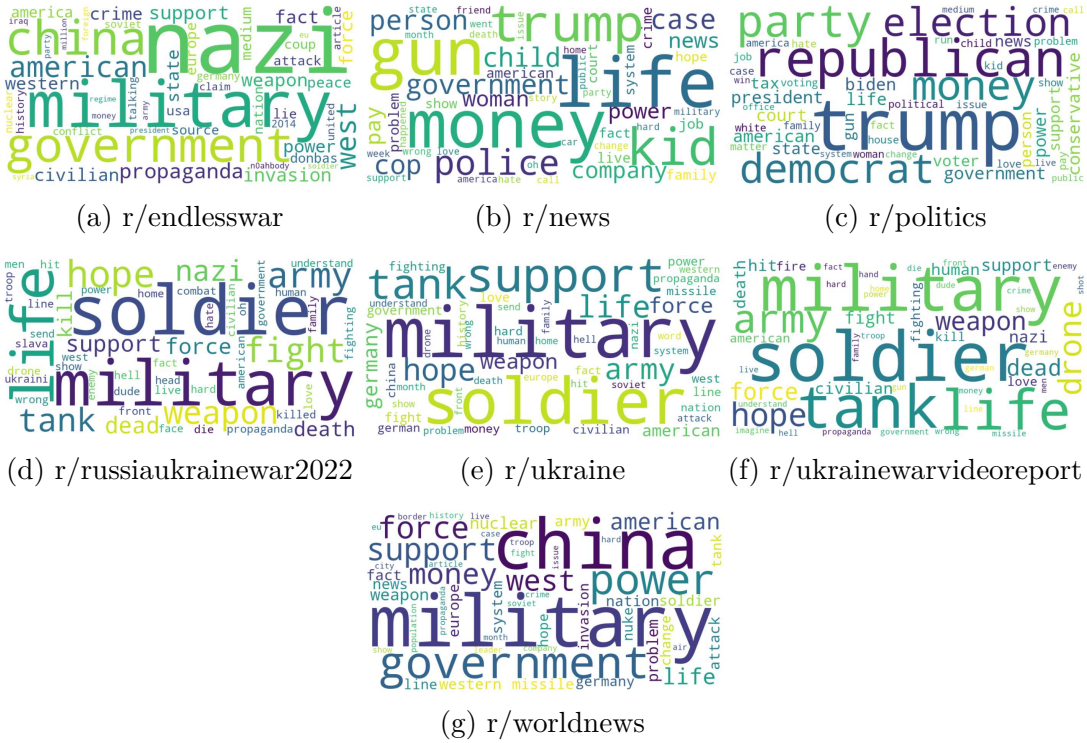
Figura 4 – Quantidade de usuários únicos por mês de conflito.



Fonte: Elaboração própria.

A seguir, na Figura 5, temos as nuvens de palavras, geradas com as 50 palavras mais comuns nas postagens e comentários de cada um dos subreddits analisados.

Figura 5 – Nuvens de palavras dos subreddits analisados.



(a) r/endlesswar (b) r/news (c) r/politics

(d) r/russiaukrainewar2022 (e) r/ukraine (f) r/ukrainewarvideoreport

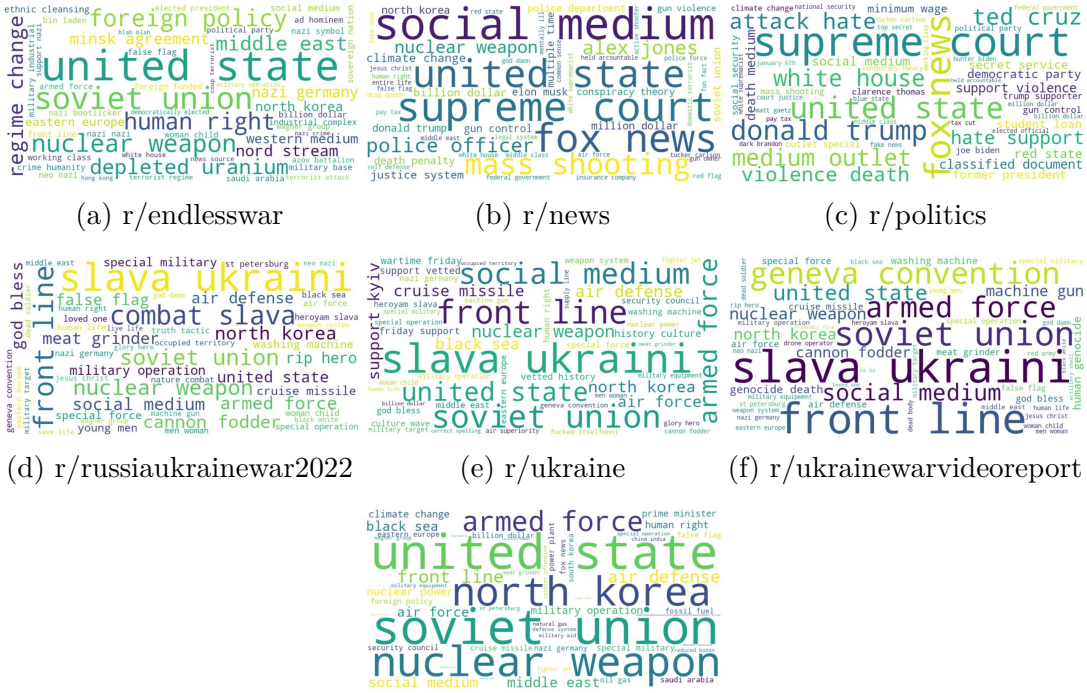
(g) r/worldnews

Fonte: Elaboração própria.

Em geral, grande parte dos termos presentes são termos militares devido à discussão em torno da guerra. Conseguimos perceber também a ocorrência de diversos termos referenciando entidades influentes na guerra, como por exemplo a China ou o ex-presidente americano Donald Trump. É possível que a presença destes termos se deva ao caráter das postagens nessas comunidades, que frequentemente são notícias sobre agentes influenciadores no conflito. A nuvem mais díspar é a do *subreddit r/politics*, que possui diversas palavras fazendo referência às eleições norte-americanas de 2020. Isso se deve ao fato de a maioria dos usuários do Reddit serem norte-americanos, juntamente ao foco político da comunidade, o que torna a discussão mais direcionada para esse contexto. Por fim, vale destacar a frequência do termo "nazi", que pode estar relacionado à grande quantidade de acusações por parte do governo russo chamando a Ucrânia de nazista.

Também é interessante analisar as nuvens de palavras dos 50 bigramas (sequências de duas palavras) mais frequentes em cada comunidade, conforme mostrado na Figura 6. Ainda há a presença de diversos termos militares, mas também surgem alguns termos mencionando mídias sociais e jornais famosos, possivelmente devido à divulgação da guerra nesses meios. O termo *soviet union* possivelmente faz referência à independência ucraniana obtida após a dissolução da União Soviética na década de 90. Além disso, vemos com frequência a menção a *slava ukraini*, que em tradução direta significa "glória à Ucrânia", uma expressão com caráter nacionalista mas que foi adotada até mesmo fora da Ucrânia como gíria de apoio ao país.

Figura 6 – Nuvens de palavras com bigramas dos subreddits analisados.



(a) r/endlesswar (b) r/news (c) r/politics

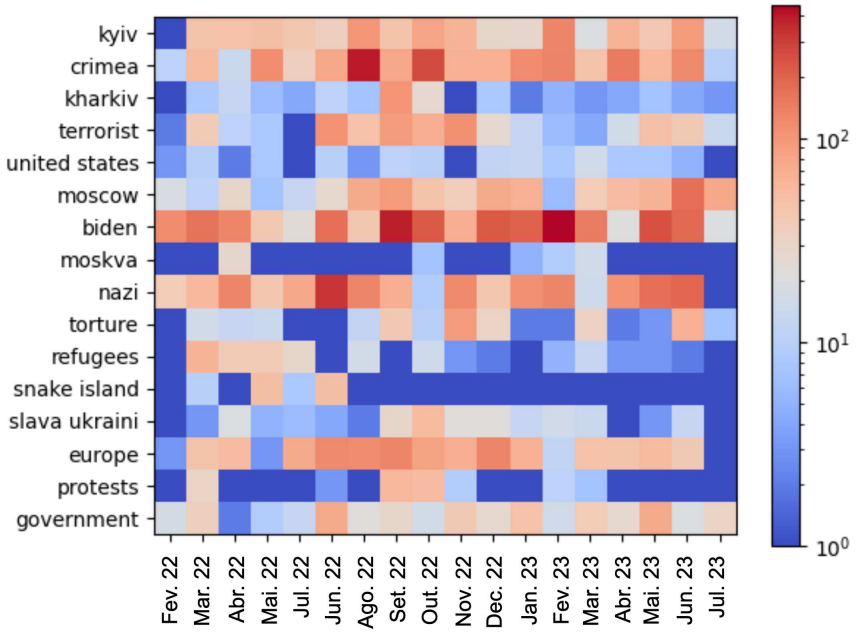
(d) r/russiaukrainewar2022 (e) r/ukraine (f) r/ukrainewarvideoreport

(g) r/worldnews

Fonte: Elaboração própria.

A Figura 7 fornece um mapa de calor, onde a cor de cada célula é representativa da quantidade de ocorrências por mês nos *subrredits* analisados de algumas palavras-chave relevantes à guerra. Nota-se uma frequência constante do termo nazi, como previamente mostrado pelas nuvens de palavra. A expressão "*slava ukraini*", por sua vez, se mostra presente nos meses de Setembro e Outubro, marcados por um grande contra-ataque da Ucrânia. Também é perceptível alguns termos que ficaram mais frequentes em momentos bem pontuais da guerra, como "*snake island*" e "*moskva*", referente a dispustas que ocorreram na Ilha das Serpentes perto do início do combate e a destruição do navio de guerra russo Moskva no mesmo período.

Figura 7 – Mapa de calor da ocorrência de palavras-chave.

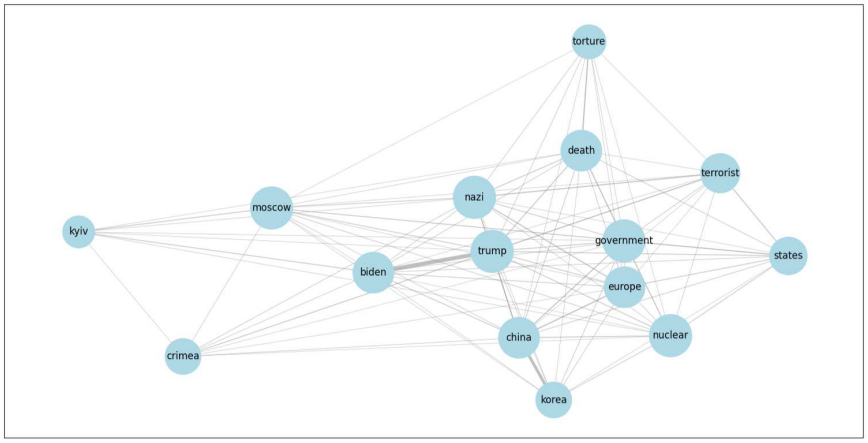


Fonte: Elaboração própria.

Outra análise interessante feita é a da rede de co-ocorrência de palavras chave, como demonstrado na Figura 8. O grafo é constituído por nodos representando palavras-chave, e arestas representando a ocorrência simultânea de duas palavras-chave. O tamanho de cada nodo é determinado pelo peso dele, ou seja, pela quantidade de arestas ligadas a ele. A grossura de cada aresta é determinada pela quantidade de vezes que o par de palavras-chave ocorreu no texto.

Todos os nodos tiveram tamanho relativamente similar, indicando que nenhuma das palavras-chave escolhidas ficou "isolada" das outras, ou seja, sem ocorrer juntamente com as outras do grafo. Podemos notar uma grande quantidade de ocorrências do par de palavras "biden" e "trump", respectivamente o atual e ex-presidente americano. A palavra "torture" (tortura) também ocorreu frequentemente junto de "death" (morte).

Figura 8 – Rede de co-ocorrência de palavras-chave.



Fonte: Elaboração própria.

4.4 ANÁLISE SEMÂNTICA DO CONJUNTO DE DADOS EXTRAÍDOS

Esta seção detalha as análises que foram feitas sobre o conjunto de dados extraídos para apresentar as diferentes opiniões e perspectivas que surgiram durante o percurso da guerra entre Rússia e Ucrânia, aplicando diferentes técnicas que serão detalhadas antes de cada análise aqui exposta.

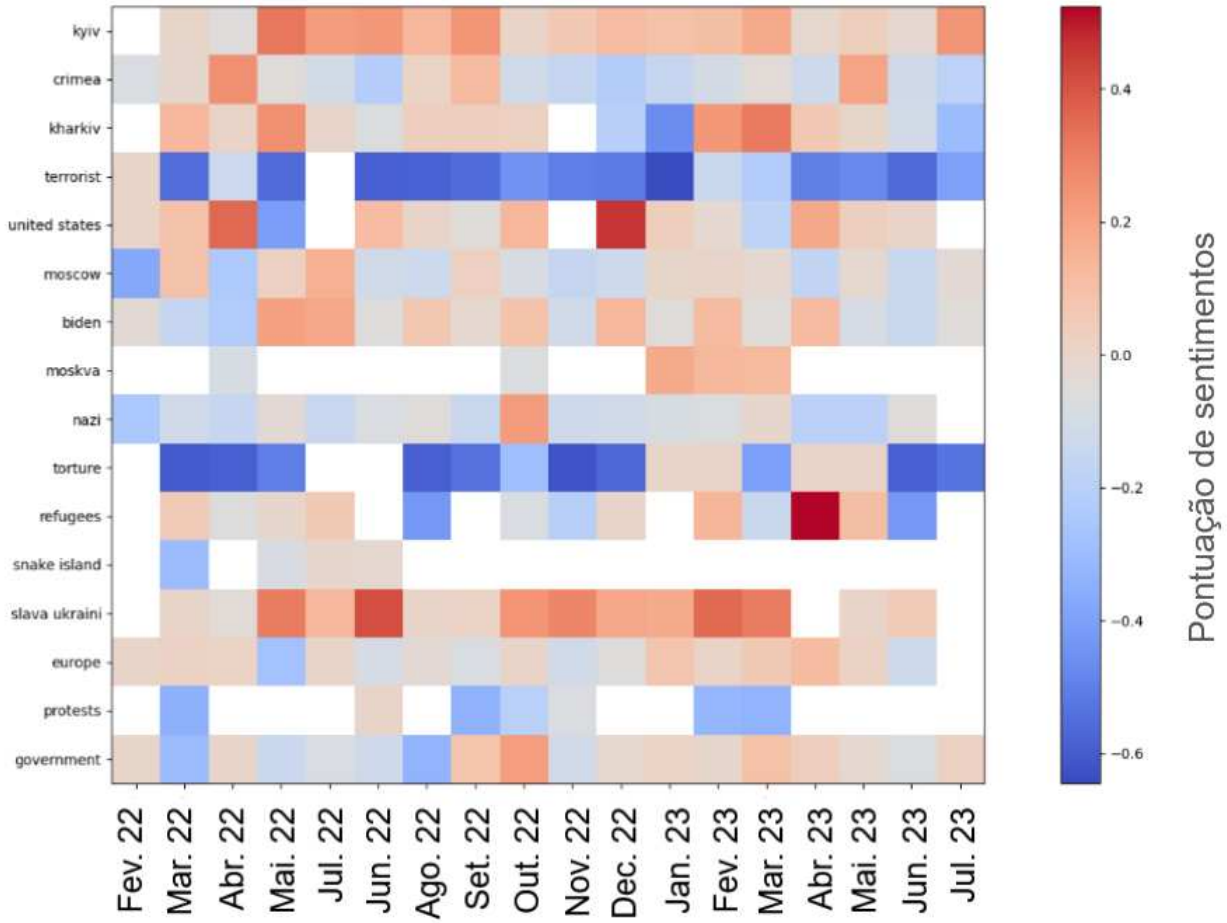
4.4.1 Análise de Sentimentos

A análise de sentimentos permite compreender melhor a atitude expressada pelos usuários em suas postagens ou comentários. Utilizamos a ferramenta VADER, usando como base o conjunto de dados de todos os *subreddits* analisados neste trabalho, para extrair uma pontuação que representa a conotação sentimental de cada interação dentro da rede social. Conforme descrito pela própria ferramenta, uma pontuação maior que 0,05 representa um sentimento positivo, enquanto uma pontuação menor que -0,05 representa um sentimento negativo.

Novamente, um conjunto de palavras relevantes da guerra foi selecionado, e a análise de sentimentos foi feita sobre as postagens e comentários que as continham, como forma de relacionar cada palavra-chave com o sentimento comumente expressado por ela no contexto das comunidades selecionadas neste trabalho. A Figura 9 mostra um mapa de calor onde cada palavra chave recebe a pontuação de sentimentos dos comentários onde ela aparece, para cada mês de guerra. Os espaços em branco representam meses onde aquela palavra não foi citada, e portanto não tem sentimento associado a ela naquele período de tempo.

Notamos que as palavras “torture” (tortura) e “terrorist” (terrorista) receberam pontuação considerada negativa em praticamente toda a extensão do gráfico, indicando que,

Figura 9 – Mapa de calor da pontuação de sentimentos de palavras-chave.



Fonte: Elaboração própria.

seja qual for o contexto em que apareceram na rede social, geralmente estavam associadas a comentários negativos. A expressão "*slava ukraini*", por sua vez, teve pontuação muito positiva em determinados momentos e neutra em outros, possivelmente por aparecer em comentários de suporte à Ucrânia independentemente de eles possuírem uma conotação mais positiva ou negativa.

Observando algumas palavras específicas, conseguimos notar a influência que certos momentos pontuais da guerra tiveram na análise de sentimentos. A menção à Ilha das Serpentes (snake island) inicialmente teve valor mais negativo, no período de invasão Russa, mas logo pende para o lado positivo nos meses de contra-ataque ucraniano. Similarmente, a palavra "Moskva" teve conotação positiva nos meses correspondente a destruição do navio russo Moskva pelo exército ucraniano.

A palavra "united states" teve pontuação bastante positiva no mês de Dezembro, marcado pelo encontro do presidente estadunidense com o presidente ucraniano. Além disso, a palavra "refugees" (refugiados) obteve pontuação geralmente neutra com alguns picos negativos, o que possivelmente nos mostra um contraste entre momentos mais

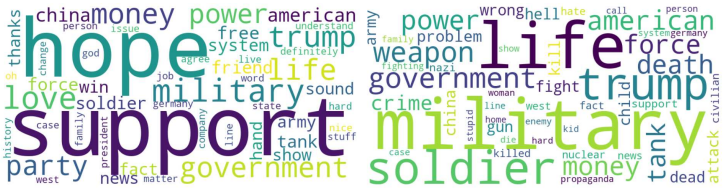
esperançosos e momentos de grandes perdas durante a guerra.

Assim, é notável como o sentimento expressado por cada palavra muda conforme a progressão da guerra, e que determinados acontecimentos tiveram grande influência nas opiniões dos usuários em momentos determinados.

4.4.2 Nuvens de palavras positivas e negativas

A partir da divisão de palavras entre positivas e negativas, baseado na pontuação obtida pela análise de sentimentos, podemos novamente utilizar nuvens de palavra como ferramenta para visualizar rapidamente as palavras normalmente associadas com sentimentos positivos ou negativos. As Figuras 10 e 11 mostram nuvens de palavras geradas utilizando especificamente conjuntos de texto que continham somente comentários positivos ou negativos, através de todos os *subreddits* analisados.

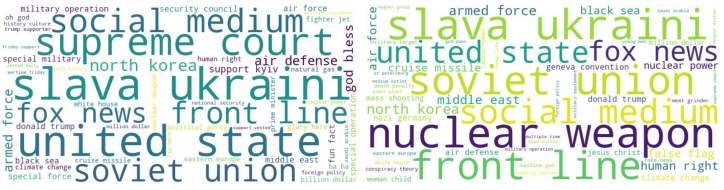
Figura 10 – Nuvens de palavras de comentários positivos e negativos.



(a) Comentários positivos (b) Comentários negativos

Fonte: Elaboração própria.

Figura 11 – Nuvens de palavras com bigramas de comentários positivos e negativos.



(a) Comentários positivos (b) Comentários negativos

Fonte: Elaboração própria.

As palavras predominantes nas nuvens de palavras positivas foram “hope” (esperança), “support” (suporte) e “slava ukraini”. A palavra “life” (vida) acabou se destacando na nuvem de palavras negativa, o que pode ser consequência do uso da palavra associado a sentimentos como lamentação ou luto. Curiosamente, muitas palavras se repetem nas nuvens de palavras negativas, como o próprio termo “slava ukraini”, indicando que comentários em geral podem ter intercalado sentimentos positivos e negativos na mesma mensagem. Por exemplo, é possível que um comentário referencie as tragédias e o caos no território ucraniano, e ainda inclua esse termo para demonstrar apoio ao país. Ainda assim, é notável que nas nuvens de palavras negativas aparecem em destaque termos militares, como “soldier” (soldado) e “nuclear weapon” (arma nuclear).

4.4.3 Análise Psicolinguística

Esta seção detalha o processo de caracterização psicolinguística do conjunto de dados, utilizando a análise LIWC (TAUSCZIK; PENNEBAKER, 2010), que caracteriza as palavras contidas nas postagens e comentários baseado no contexto em que elas aparecem. Essa técnica nos permite identificar melhor a conotação emocional de cada palavra, além de identificar se uma palavra está associada a algum tema específico, como por exemplo família, trabalho etc.

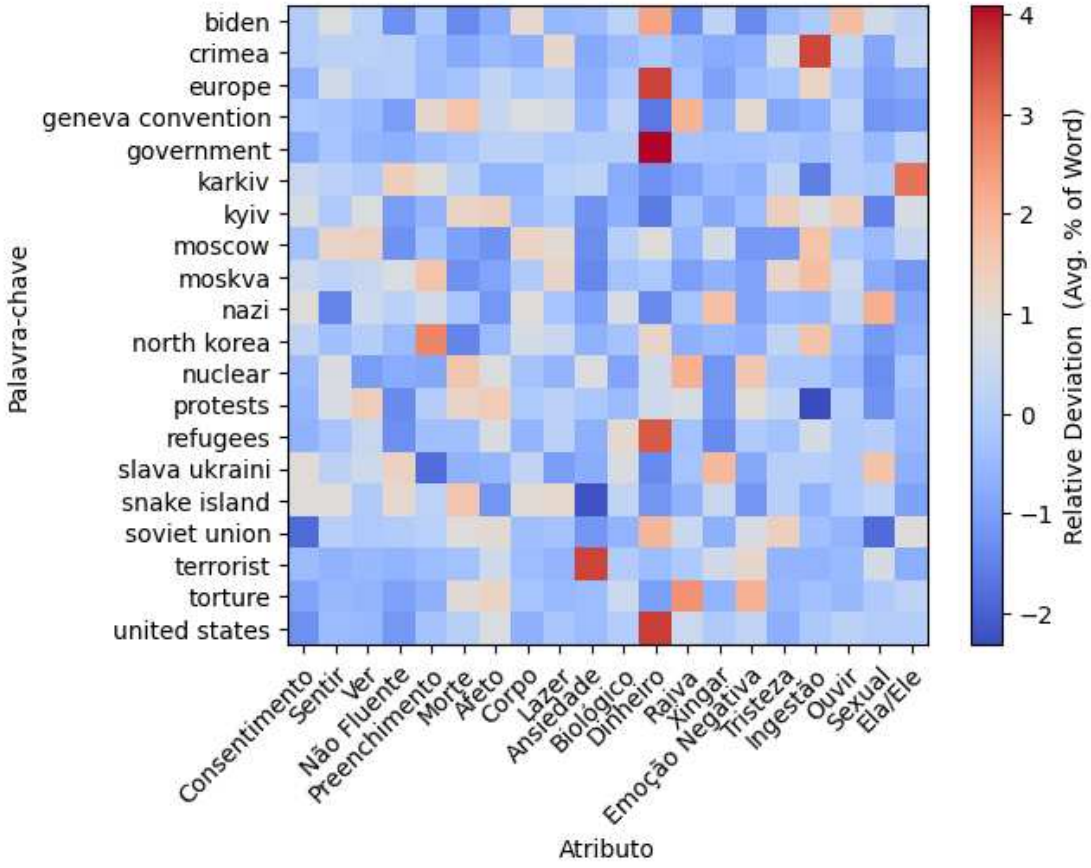
A análise inicialmente detectou 64 categorias, ou atributos, que estavam presentes no conjunto de dados. Destes 64, aqueles que possuem diferenças significativas entre as palavras-chave podem ser identificados por meio do teste de Kruskal-Wallis (KRUSKAL; WALLIS, 1952). No caso aqui analisado, todos os atributos foram consideravelmente diferentes, então foi necessário aplicar o coeficiente de Gini (CATALANO; LEISE; PFAFF, 2009) para identificar quais eram mais discriminantes, e assim limitar a grande quantidade de atributos identificados.

O mapa de calor na Figura 12 mostra os 20 atributos mais discriminantes encontrados pelo coeficiente de Gini. A cor de cada célula é determinada pelo desvio relativo da palavra-chave sobre cada atributo, quando comparada com as outras palavras-chave. Assim, uma célula fica mais vermelha se o desvio for acima da média, e mais azul se o desvio for abaixo da média. De modo geral, quando uma palavra-chave tem um desvio alto relacionado a um atributo, significa que postagens e comentários que possuem aquela palavra-chave fazem uso de outros termos relacionados ao atributo em destaque.

Notamos que diversas palavras-chave ficaram relacionadas ao atributo "dinheiro", como "*united states*", "*refugees*", "*europe*", "*government*" e "*biden*". O atributo morte também mostrou relação com as palavras "*torture*", "*nuclear*", "*geneva convention*" e "*kyiv*". O atributo de ansiedade ficou relacionado especificamente à palavra "*terrorist*", que se destacou na coluna comparada a todas as outras palavras.

A análise LIWC serve como complemento para compreender melhor o contexto onde cada palavra se encaixa, e é útil como ferramenta auxiliar, juntamente com as outras análises descritas nesta seção, para identificar a presença de comunidades definidas por opiniões semelhantes entre si na segunda etapa deste trabalho.

Figura 12 – Mapa de calor representando os atributos mais discriminantes da análise LIWC.



Fonte: Elaboração própria.

5 ANÁLISE DE GRAFOS E COMUNIDADES

Este capítulo tem como foco a apresentação dos grafos gerados a partir dos dados coletados de cada *subreddit* e posteriormente o processo de detecção de Comunidades dentro destes gráficos utilizando diferentes algoritmos. A partir disso, é realizada a análise dos atributos de cada grafo e das Comunidades identificadas, correlacionando os resultados obtidos com o contexto da rede social Reddit.

5.1 GERAÇÃO DOS GRAFOS

Os grafos foram gerados com auxílio da biblioteca NetworkX[1] em Python. Cada nodo dentro destes grafos representa um usuário, e cada aresta uma interação entre dois usuários. Além disso, as arestas destas redes geradas para cada subreddit são dirigidas, de forma que se o usuário A mandou um comentário para o usuário B, a aresta terá direção de A para B. Vale ressaltar que usuários podem mandar mensagens para eles mesmos, gerando arestas que fazem um *loop* para o nodo de onde saem. Caso existam múltiplas interações de um mesmo caminho A para B, o peso da aresta é incrementado em 1 para cada interação.

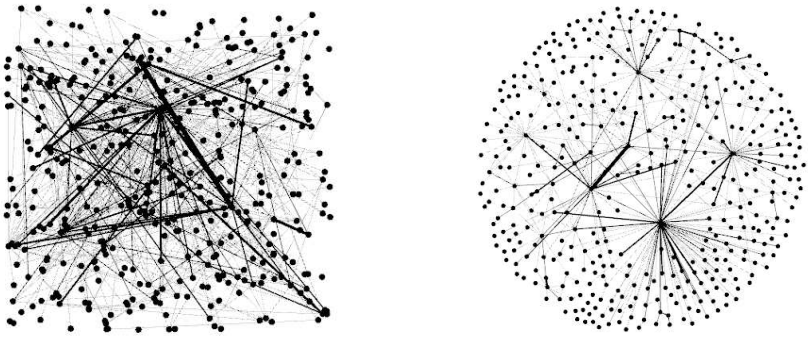
Os grafos gerados puderam ser visualizados com a utilização da ferramenta Gephi[2]. O Gephi disponibiliza alguns algoritmos convenientes para reorganizar a posição dos nodos e arestas da rede, de forma que sua visualização fique mais clara e de fácil compreensão. Com isso, é possível extrair algumas estatísticas capazes de expandir o conhecimento sobre as redes de interação de usuários representadas pelos grafos. Além disso, esse processo permite comparar as estatísticas entre grafos de interação, para entender as diferenças entre cada *subreddit*.

O algoritmo usado para organizar os grafos neste trabalho foi o Force Atlas 2, disponível dentro do Gephi por padrão. Este algoritmo proporciona uma distribuição baseada em repulsão de nodos e atração de arestas (JACOMY *et al.*, 2014), de forma que após algum tempo da execução, a rede alcança um estado de equilíbrio onde nodos que estão mais conectados ficam próximos, e nodos menos conectados ficam mais distantes.

A Figura 13 mostra como um grafo qualquer é representado visualmente no Gephi antes e depois da execução do algoritmo de reorganização.

[1] Disponível em: `https://networkx.org/`
[2] Disponível em: `https://gephi.org/`

Figura 13 – Um grafo de exemplo, antes e depois da execução do algoritmo Force Atlas 2.



(a) Antes (b) Depois

Fonte: Elaboração própria.

5.2 CARACTERIZAÇÃO DOS GRAFOS

Os dados coletados de todos os *subreddits* tema deste trabalho foram usados para gerar um grafo referente a cada *subreddit*. Estes grafos são uma combinação dos dados de todo o período que a coleta abrangeu, de forma que cada uma dessas redes representam os usuários que participaram de cada *subreddit* dentro do período, juntamente com as interações (comentários) que ocorreram entre estes usuários.

A Tabela 3 mostra as estatísticas encontradas para os grafos de todos os *subreddits* analisados. O diâmetro da rede é a maior distância entre dois nodos quaisquer da rede, considerando que nodos vizinhos tem 1 de distância. A densidade é uma medida de completude do grafo, de forma que um grafo onde todos os nodos tem arestas que os conectam entre si teria densidade 1.

Subreddit	Nº de Nós	Nº de Arestas	Maior Grau	Grau médio	Diâmetro
r/EndlessWar	2581	12105	646	4,69	9
r/News	125862	275598	1692	2,19	27
r/Politics	99867	230426	2513	2,307	22
r/RussiaUkraineWar2022	31546	124361	3215	3,942	15
r/Ukraine	103289	431291	3680	4,176	23
r/UkraineWarVideoReport	85219	311319	6033	3,653	18
r/WorldNews	183845	437223	2720	2,37	20

Tabela 3 – Tabela de dados encontrados para os grafos de todos os *subreddits*.

Fonte: Elaboração própria.

5.2.1 r/EndlessWar

A Tabela 2 nos fornece algumas estatísticas do grafo do *subreddit* “r/EndlessWar”. Em relação ao grau médio, é interessante que se considere a distribuição do grau através dos nodos além de seu valor. Assim, a Figura 14 mostra um gráfico que contém essa

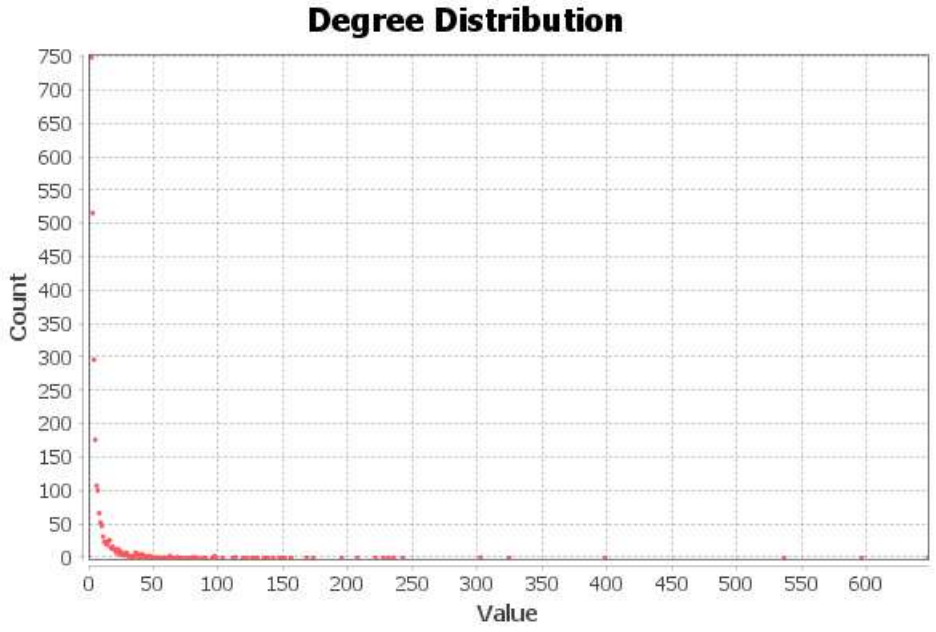
distribuição, indicando pelo eixo "x" o valor de grau de cada nodo, e pelo eixo "y" a quantidade de nodos que possuem aquele valor de grau.

Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
2581	12105	646	4,69	9

Tabela 4 – Tabela de dados do grafo dirigido de r/EndlessWar

Fonte: Elaboração própria.

Figura 14 – Distribuição do grau dos nodos do grafo de r/EndlessWar.



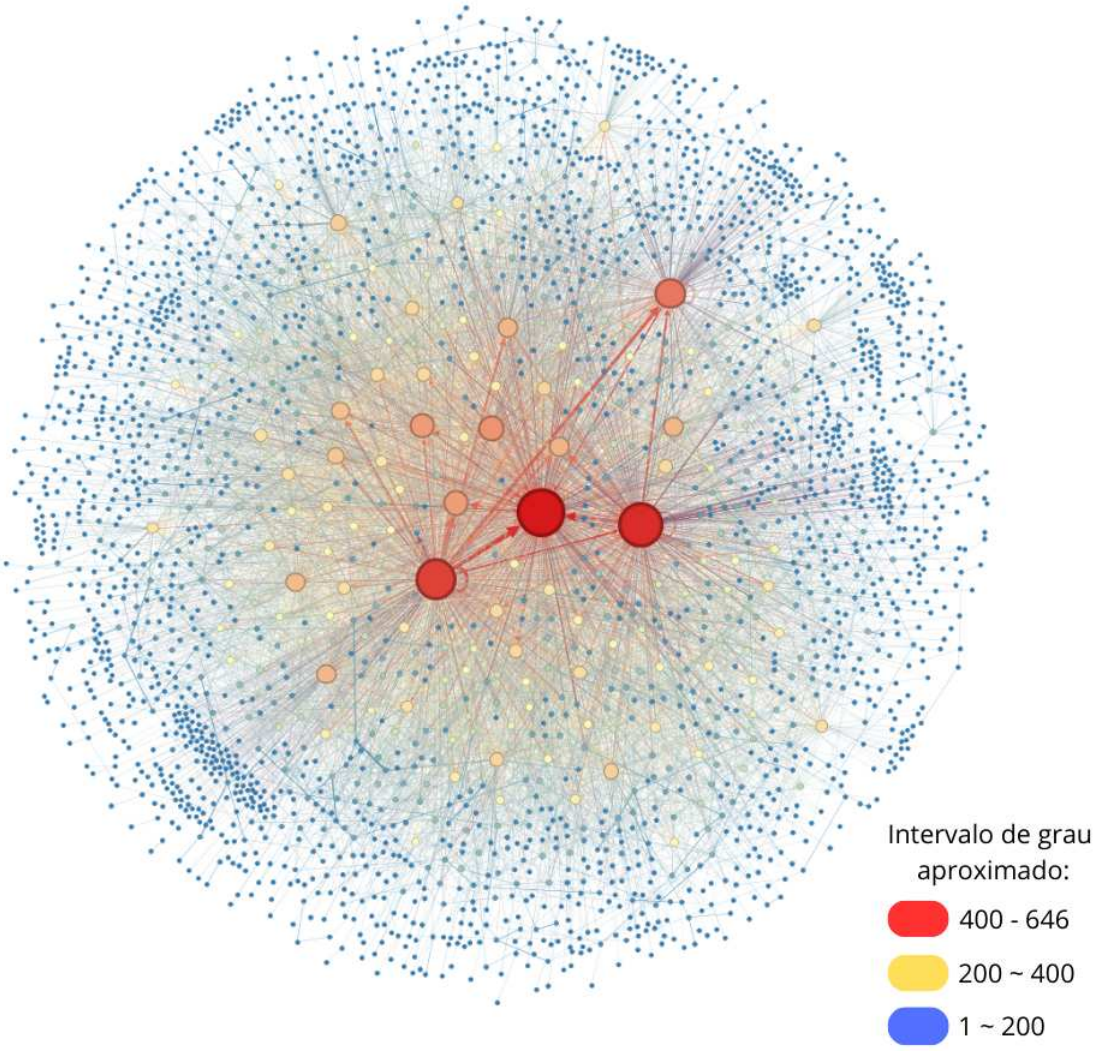
Fonte: Elaboração própria.

Destas estatísticas, é possível extrair algumas informações do contexto do fórum da rede social. Primeiramente, observando o grau médio, é notável que alguns poucos nodos possuem um grau bem elevado comparado com o restante. É possível que isso seja consequência da estrutura dos tópicos de discussão no Reddit, onde comentários e postagens que possuem muitos *upvotes* são levados para o início da página, enquanto comentários com menos *upvotes* ficam no final do fórum. Assim, existe uma tendência de que usuários que foram "arrastados" para o início da página tenham uma quantidade extremamente alta de interações, aumentando bastante o grau de seu nodo correspondente no grafo, em contrapartida a comentários com poucos votos, que dificilmente receberão alguma resposta.

A Figura 15 mostra a visualização do grafo gerado a partir dos dados coletados do *subreddit r/endlesswar*. Este grafo é dirigido, indicando a origem e o destino de cada comentário por meio das setas nas pontas das arestas. Além disso, alguns elementos da

visualização foram modificados para facilitar a compreensão. Primeiramente, o tamanho dos nodos cresce de acordo com seu grau total (soma da quantidade de arestas saindo e quantidade de arestas entrando no nodo). Similarmente, as arestas ficam maiores com base em seu peso (quantas interações ela representa). Por fim, o grafo foi colorido também baseando-se no grau dos nodos, onde pontos azulados representam um grau pequeno, enquanto pontos mais alaranjados ou avermelhados representam um grau relativamente alto comparado com o restante da rede.

Figura 15 – Grafo dirigido dos dados coletados de r/EndlessWar.



Fonte: Elaboração própria.

O grafo possui diversos pontos de alta densidade (*hubs*), representados pelos nodos mais avermelhados e maiores, que notavelmente possuem uma grande quantidade de arestas conectadas a eles. Isso indica usuários que enviaram muitas respostas para outros comentários, receberam muitas respostas em seus comentários, ou ambos. Como este *subreddit* em específico tem um tamanho pequeno comparado aos outros analisados neste

trabalho, também é possível que esses usuários possuam alguma relação direta com o fórum, como por exemplo o cargo de moderador, e consequentemente interagem de forma frequente o suficiente para influenciar o tamanho de seus respectivos nodos no grafo.

Como mencionado anteriormente, o algoritmo Force Atlas 2 tenta manter próximos na visualização nodos que estejam mais conectados entre si. Isso significa que a posição de cada nodo também pode fornecer informação relevante. Neste caso, nodos que ficaram mais próximos da borda do grafo podem indicar usuários que não interagem tão frequentemente no *subreddit*, participando de apenas algumas discussões seletas (o que não impede que o usuário tenha uma grande quantidade de interações dentro dessas discussões). Nodos que estão mais próximos do meio do grafo, por sua vez, podem indicar usuários que interagiram em variados tópicos de discussão através de diferentes postagens na Comunidade, o que significa que receberam ou enviaram comentários à uma grande variedade de usuários.

Neste *subreddit*, também é perceptível uma grande quantidade de nodos amarelados ou alaranjados perto das bordas do grafo, que recebem arestas de um pequeno grupo de pontos azulados pequenos. Esse fenômeno pode indicar usuários que alcançaram repercussão em alguma postagem ou comentário pontual, mas não necessariamente são frequentadores da Comunidade.

5.2.2 r/News

A Tabela 5 informa as estatísticas do grafo gerado a partir de dados do *subreddit* "r/News", e a Figura 16 mostra sua distribuição de grau. Além disso, a Figura 17 mostra sua visualização.

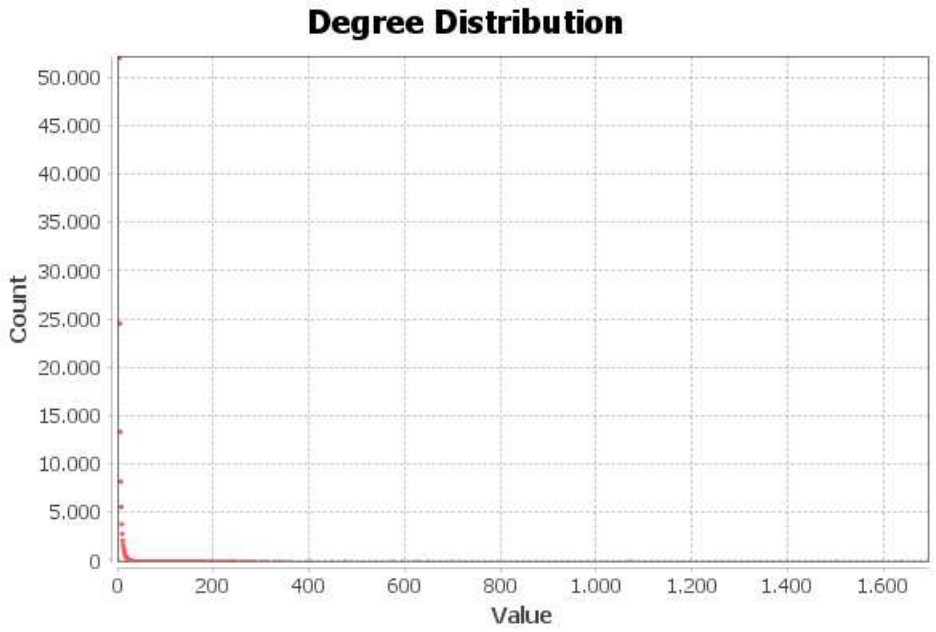
Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
125862	275598	1692	2, 19	27

Tabela 5 – Tabela de dados do grafo dirigido de r/News

Fonte: Elaboração própria.

Esse *subreddit* é uma Comunidade bem grande, não apenas relativo às outras analisadas neste trabalho, mas também à rede social Reddit como um todo. Dessa forma, é notável que o grafo fica bem denso. Uma característica importante de ser comentada aqui é que Comunidades que são grandes o suficiente dentro do Reddit geralmente recebem destaque na página inicial do *site*, aparecendo até para usuários que nunca a acessaram ou que não sabem previamente de sua existência. Dessa forma, é natural que esses *subreddits* destacados recebam um influxo de usuários vindos da página inicial, que interagirão dentro do fórum mas não necessariamente se tornarão frequentadores ou ao menos permanecerão muito tempo nele. Esse contexto pode ser uma explicação para a grande quantidade de nodos com grau relativamente baixo que preenchem este grafo. Naturalmente, isso

Figura 16 – Distribuição do grau dos nodos do grafo de r/News.



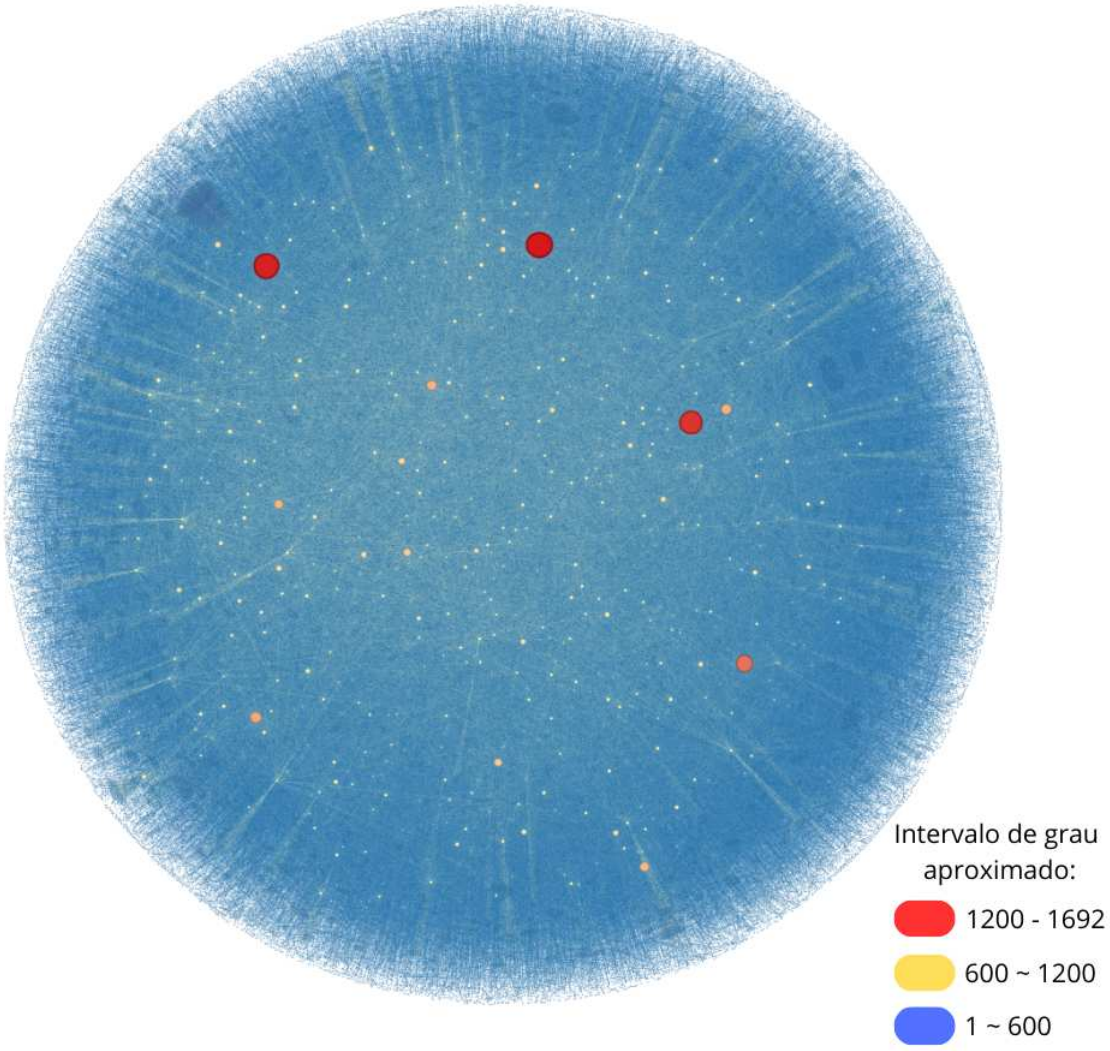
Fonte: Elaboração própria.

também justifica uma média de grau menor, apesar da presença de alguns nodos com grau excepcionalmente alto.

Os nodos mais amarelados, representando um valor de grau mediano relativo aos outros, estão bem distribuídos através de todo o grafo. Este intervalo de grau pode acontecer devido a uma grande variedade de fatores. Para os nodos mais próximos da borda, é possível que representem usuários que obtiveram uma repercussão considerável em suas postagens e comentários, mas apenas em momentos pontuais dentro do período coletado, e por isso não estão tão conectados com o resto do grafo. Os nodos mais próximos do centro, por sua vez, podem representar usuários que são frequentadores do *subreddit* e por isso estão ligados a muitos outros, e interagem constantemente.

Os nodos maiores, ou *hubs*, podem indicar usuários que não apenas interagem constantemente na Comunidade, mas obtém uma grande repercussão em suas postagens e comentários. Vale ressaltar que o Reddit é uma rede social menos centrada em usuários e mais no conteúdo que eles publicam, o que implica que esses *hubs* não necessariamente são usuários famosos ou particularmente importantes dentro da rede, mas que o conteúdo que produzem recebe grande reconhecimento dentro deste *subreddit*, seja por relevância, qualidade ou algum outro fator.

Figura 17 – Grafo dirigido dos dados coletados de r/News.



Fonte: Elaboração própria.

5.2.3 r/Politics

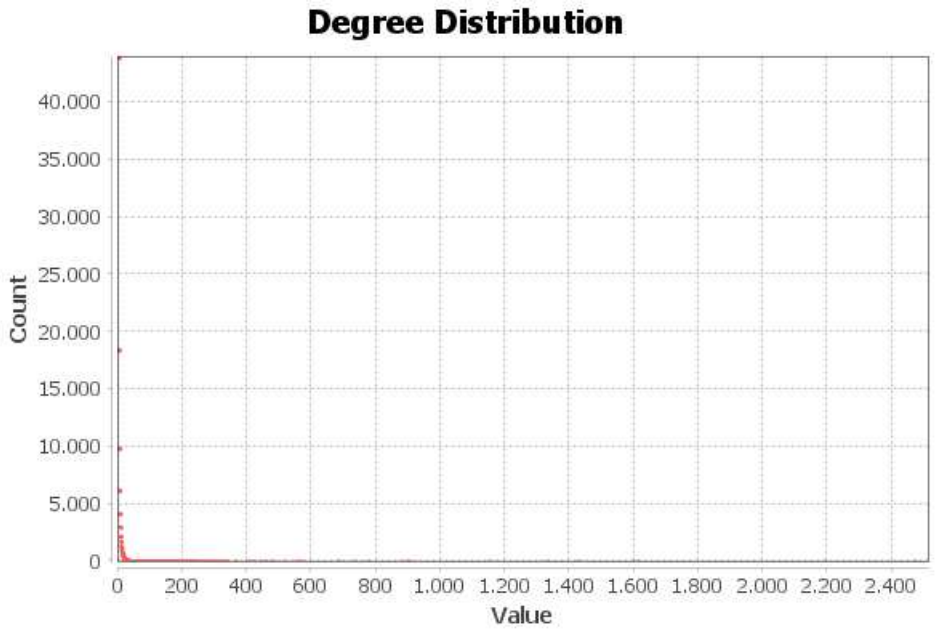
O *subreddit* "r/Politics" é estruturalmente parecido com "r/News", por também ser um fórum grande e que frequentemente recebe destaque na página inicial do *site*. A Tabela 6 que fornece as estatísticas de seu grafo, a Figura 18 com a distribuição de grau, e a visualização do grafo na Figura 19

Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
99867	230426	2513	2, 307	22

Tabela 6 – Tabela de dados do grafo dirigido de r/Politics

Fonte: Elaboração própria.

Figura 18 – Distribuição do grau dos nodos do grafo de r/Politics.

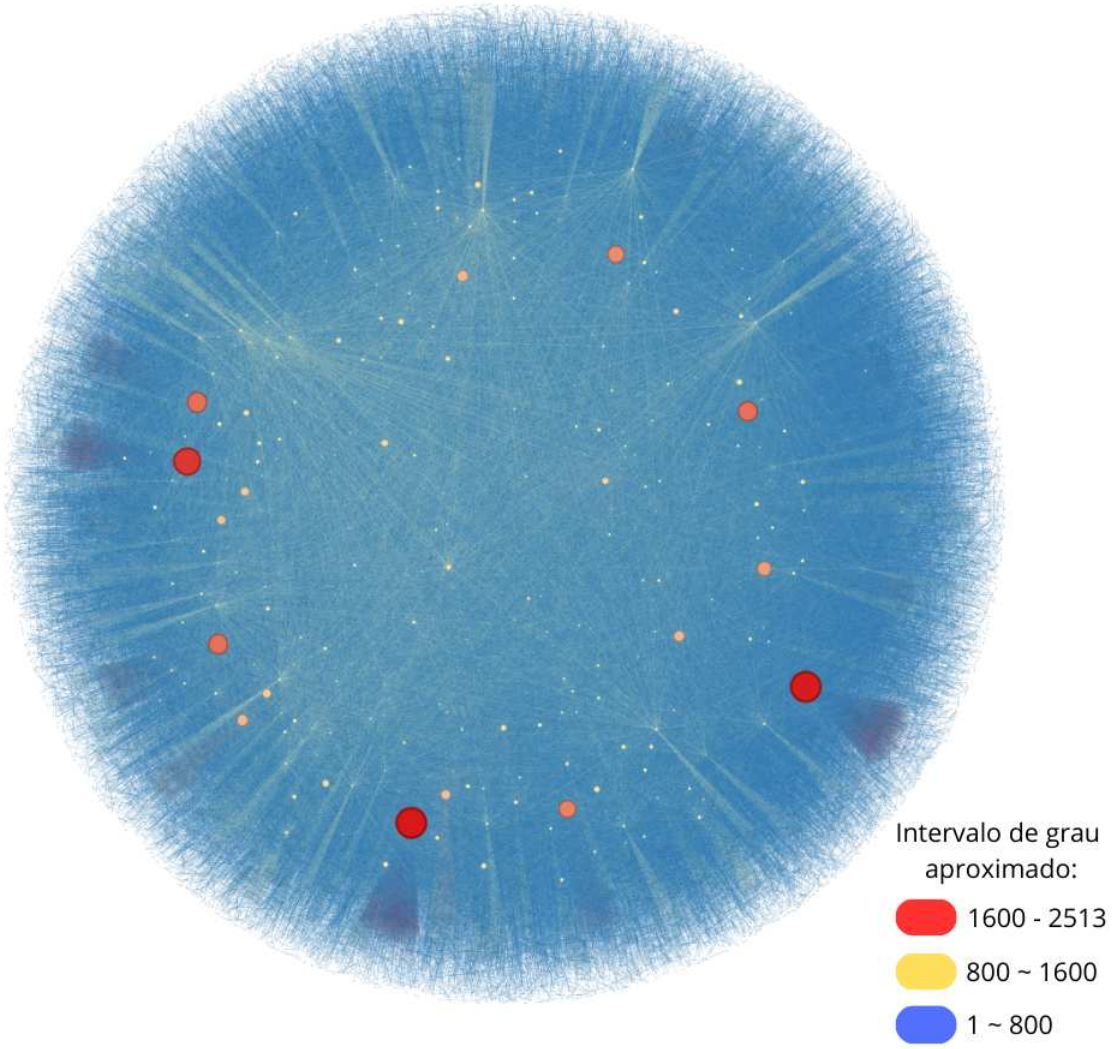


Fonte: Elaboração própria.

É perceptível que, comparado ao grafo anterior, este grafo tem menos *hubs* perto do centro. Considerando o contexto político da Comunidade, isso pode indicar que as interações conectadas aos *hubs* aconteceram principalmente em momentos com grande impacto político durante a guerra, onde ocorreram interações entre usuários que não estão tão presentes no fórum durante o resto do período de coleta. Neste grafo, também são bem perceptíveis diversos focos de nodos de grau relativamente menor contornando a rede. Isso também pode ser sinal de que o *subreddit* tem um influxo de usuários que participaram da discussão poucas vezes e provavelmente não frequentaram mais estes tópicos.

Outro fenômeno que se repete bastante neste grafo pode ser observado pela disposição das arestas. Observando atentamente, é notável a frequência com que aparecem nodos amarelados ou avermelhados perto das bordas do grafo, que recebem uma grande quantidade de arestas vindas de nodos menores, ainda mais próximos do limite da rede. Isso geralmente representa, no contexto de uma discussão no Reddit, uma postagem ou comentário com grande quantidade de réplicas, mas pequena ou nenhuma quantidade de tréplicas (quando um usuário responde "de volta" um comentário que recebeu). Essa situação pode implicar, por exemplo, um comentário que foi controverso, ou simplesmente um tópico que não rendeu nenhum debate mais aprofundado.

Figura 19 – Grafo dirigido dos dados coletados de r/Politics.



Fonte: Elaboração própria.

5.2.4 r/RussiaUkraineWar2022

As estatísticas deste grafo estão presentes na Tabela 7, seguidas do gráfico de distribuição de grau (20) e visualização (21). Um ponto curioso deste grafo, é que apesar do número muito menor de nodos comparado a alguns outros subreddits, o maior grau foi muito elevado. Isso indica que, em algum momento da Comunidade dentro do intervalo de tempo dos dados coletados, o usuário representado pelo maior nodo do grafo, destacado em vermelho, obteve muita repercussão em suas postagens ou comentários.

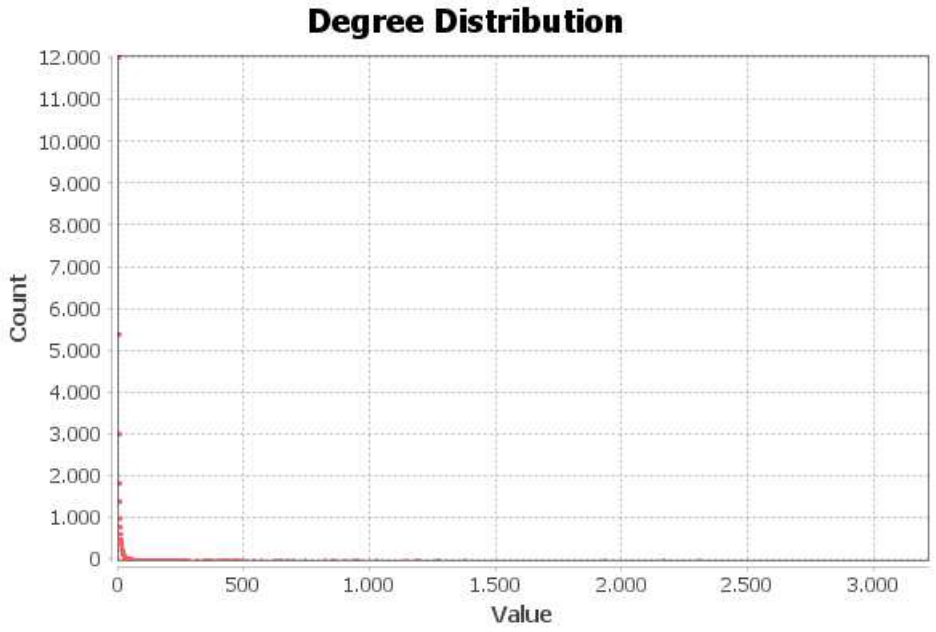
Vale notar que, diferentemente dos *subreddits* anteriores, este foi criado e é dedicado especificamente para a guerra da Ucrânia. Uma possível consequência disso é que usuários participantes desta Comunidade estão interessados apenas no assunto da guerra, diferentemente dos outros *subreddits* que tinham postagens não relacionadas. Assim, é possível que os membros dessa Comunidade sejam mais "fiéis" a ela, de forma que ativamente

Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
31546	124361	3215	3,942	15

Tabela 7 – Tabela de dados do grafo dirigido de r/RussiaUkraineWar2022

Fonte: Elaboração própria.

Figura 20 – Distribuição do grau dos nodos do grafo de r/RussiaUkraineWar2022.

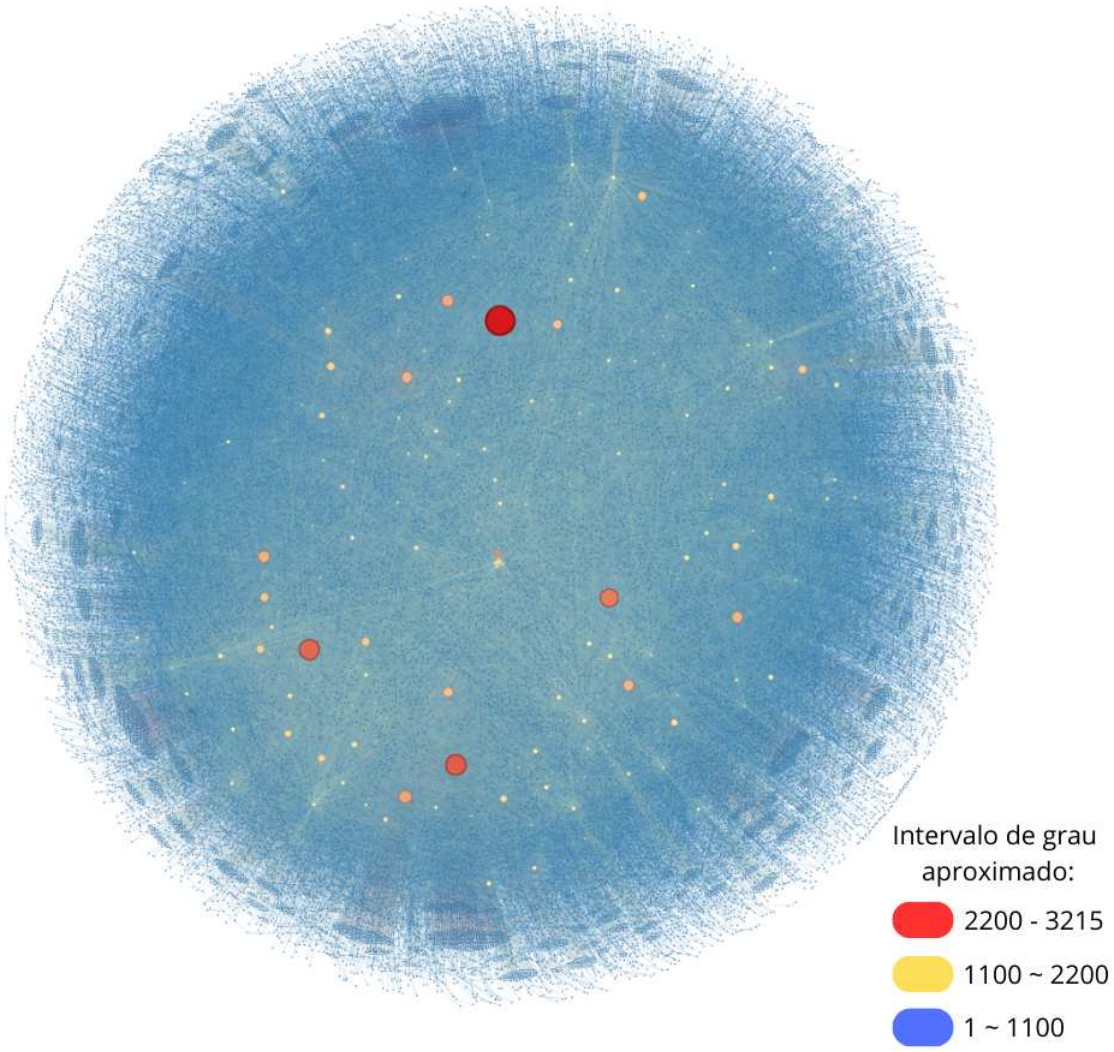


Fonte: Elaboração própria.

buscam interagir e acompanhar as discussões a medida que a guerra se desenrola. Essa situação é refletida neste grafo, onde a presença de nodos com grau mais elevado ficou bem distribuída através de toda a sua extensão.

A distribuição dos nodos de grau mediano também denota uma região do grafo com grande quantidade de interações. É perceptível a concentração desses nodos na área central-inferior, possivelmente indicando que estes usuários interagiram muito frequentemente entre si. Analisando o gráfico de distribuição de grau, esta Comunidade parece ter valores de grau (eixo "x") distribuidos de forma mais constante que os outros *subreddits* analisados.

Figura 21 – Grafo dirigido dos dados coletados de r/RussiaUkraineWar2022.



Fonte: Elaboração própria.

5.2.5 r/Ukraine

Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
103289	431291	3680	4, 176	23

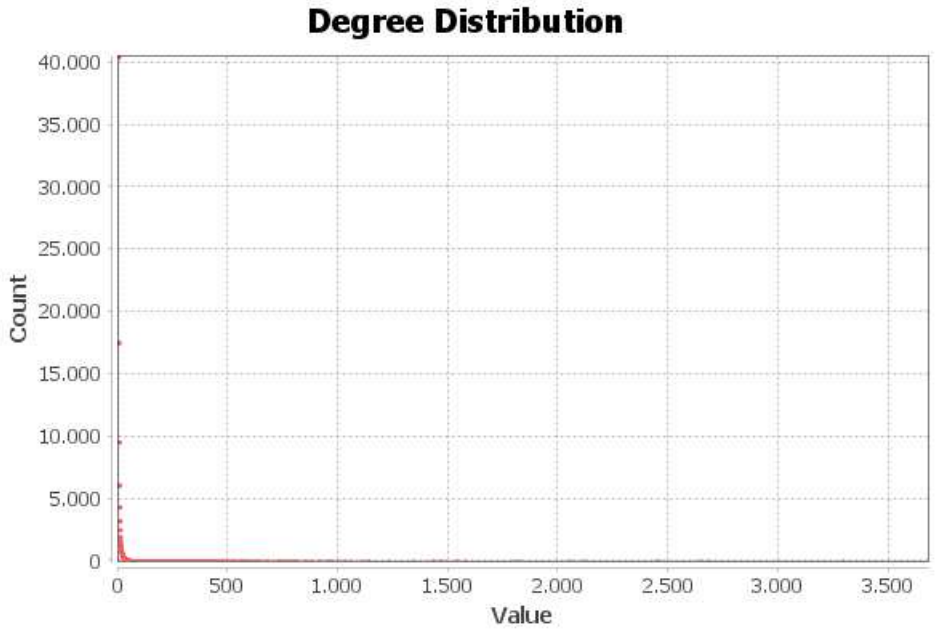
Tabela 8 – Tabela de dados do grafo dirigido de r/Ukraine

Fonte: Elaboração própria.

Esse *subreddit*, como mostra a Tabela 8, tem uma quantidade de arestas altíssima comparado com os outros analisados neste trabalho. De fato, observando a visualização do grafo na Figura 23, fica claro a grande densidade de interações que ocorrem durante o período de coleta dos dados neste fórum. É importante explicar aqui o contexto desta Comunidade dentro da rede social para analisar este grafo. O *subreddit* "r/Ukraine",

desde o início da guerra se tornou um grande centro de discussão e compartilhamento de informação sobre ela, não apenas no Reddit como também na internet como um todo. Essa página recebeu muitos papéis diferentes com o conflito, dentre eles: *feed* de notícias relacionadas a guerra, local de mobilizações sociais para apoio aos ucranianos, fórum de discussão política, militar e humanitária etc. Possivelmente, foram essas as razões que causaram uma concentração tão grande de interações neste grafo, comparado aos outros neste trabalho.

Figura 22 – Distribuição do grau dos nodos do grafo de r/Ukraine.

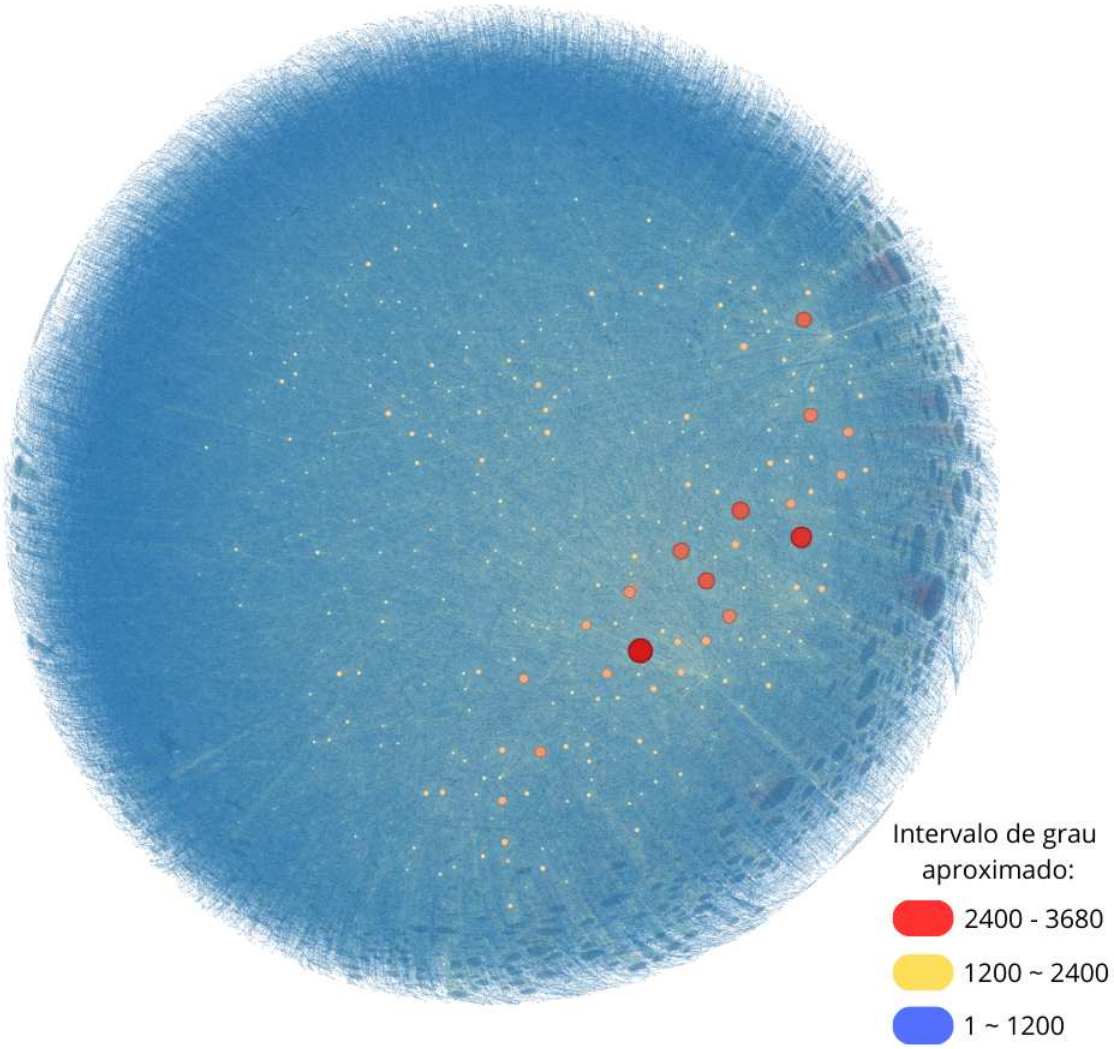


Fonte: Elaboração própria.

É perceptível que a maioria dos *hubs* ficaram na mesma região do grafo, mais para a metade direita. Baseando-se na ideia de que usuários frequentadores do fórum no mesmo intervalo de tempo tem maior probabilidade de se conectarem, podemos inferir que essa concentração de nodos que ocorre no grafo pode ter acontecido devido ao imenso fluxo de usuários novos para a Comunidade (que anteriormente a guerra servia mais como um local direcionado para ucranianos) quando o conflito começou, que posteriormente deixaram de frequentar as discussões e por isso não criaram conexões através de toda a extensão do grafo. Nodos amarelados que ficaram mais para o centro, longe dos *hubs* principais, podem indicar usuários frequentadores do fórum durante um intervalo de tempo mais extenso, de forma a terem uma grande quantidade de interações mas não estarem tão conectados com os pontos de maior concentração.

Por fim, também são marcantes as conglomerações de nodos com grau mais baixo perto das bordas do grafo. Esses nodos representam grupos de usuários que enviaram interações para os *hubs*, mas não foram respondidos de volta. Esse fenômeno é comum

Figura 23 – Grafo dirigido dos dados coletados de r/Ukraine.



Fonte: Elaboração própria.

no Reddit quando, por exemplo, uma postagem ou comentário de um usuário gera muita repercussão, de forma que o usuário não tem capacidade de replicar todo o contato que recebe. Além disso, esses agrupamentos também podem indicar usuários que interagiram com a postagem ou comentário publicado por um nodo de grau alto muito tempo depois que a publicação aconteceu, e por isso não foram respondidos.

5.2.6 r/UkraineWarVideoReport

Esse *subreddit* também foi criado após o início da guerra, e seu foco é especificamente no compartilhamento de video e imagens relacionadas a guerra. Esse conteúdo varia desde notícias e relatórios até registros explícitos da violência no conflito. Postagens desse tipo podem provocar bastante repercussão por parte dos usuários, devido ao choque e à revolta causados pela exposição a material sensível. Esse contexto pode ser uma explicação para

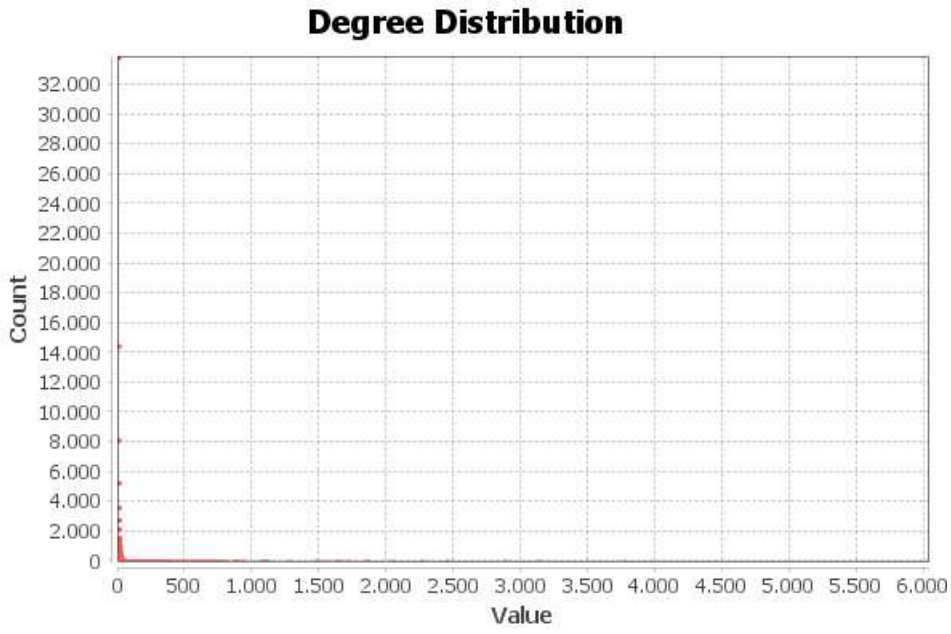
o grande número de interações, indicadas pelas arestas na Tabela 9. Similarmente, o grau máximo atingido também foi de um valor relativamente alto comparado aos outros grafos analisados aqui, indicando um usuário que atingiu muita repercussão em suas postagens e comentários.

Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
85219	311319	6033	3,653	18

Tabela 9 – Tabela de dados do grafo dirigido de r/UkraineWarVideoReport

Fonte: Elaboração própria.

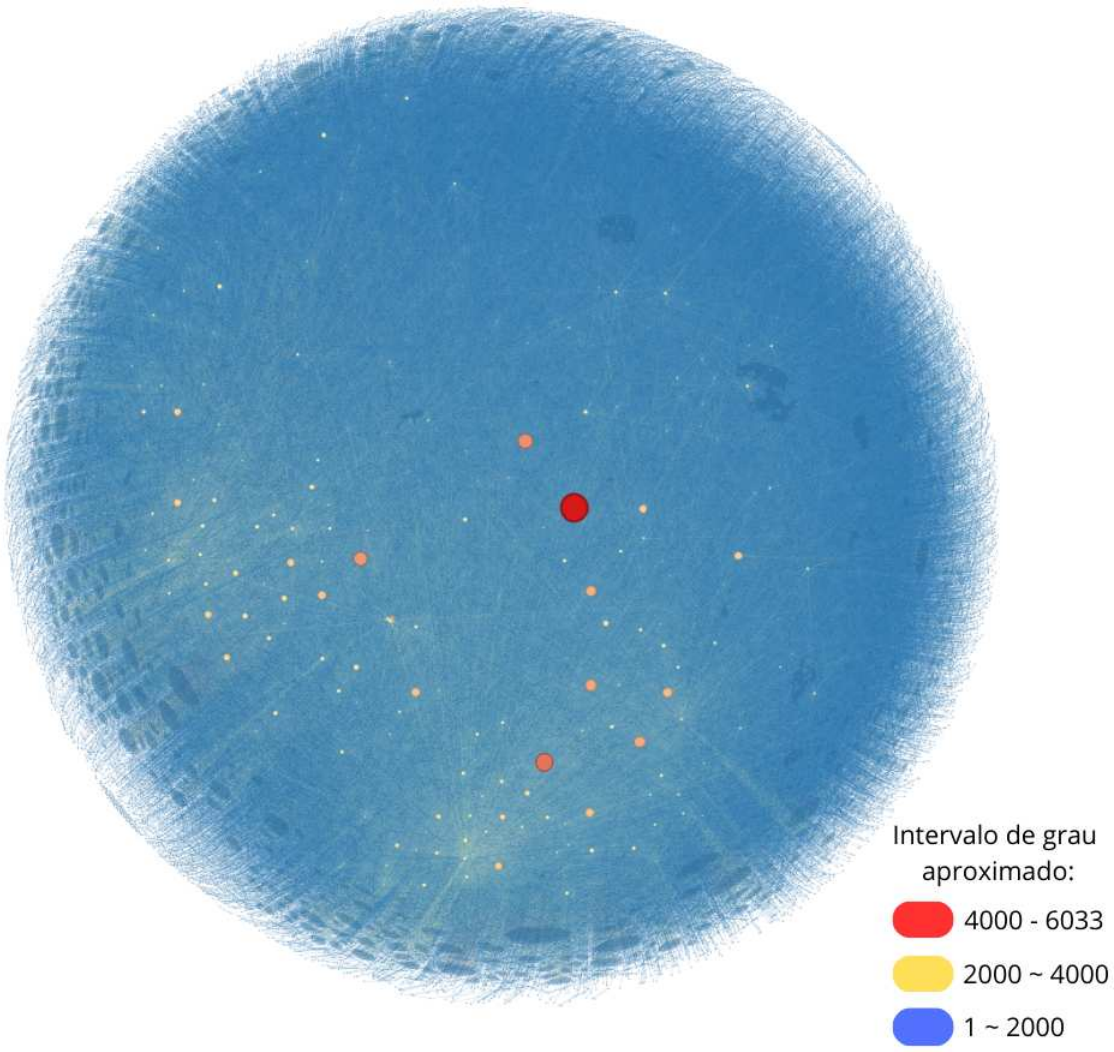
Figura 24 – Distribuição do grau dos nodos do grafo de r/UkraineWarVideoReport.



Fonte: Elaboração própria.

Analisando a distribuição de grau na Figura 24 e a visualização do grafo na Figura 25, vemos que o grafo no geral não tem muitos nodos no intervalo de grau mais elevado, apesar de os *hubs* terem atingido valores de grau bem altos. Uma possível explicação é pelo fato de as postagens do *subreddit* serem, em sua maior parte, compostas por vídeos e imagens. Nesse cenário, cada postagem pode receber uma grande quantidade de respostas (principalmente se o conteúdo exposto nelas for chocante) mas ao mesmo tempo não provocar muita discussão (réplicas e tréplicas) entre os usuários, tornando a rede menos densa.

Figura 25 – Grafo dirigido dos dados coletados de r/UkraineWarVideoReport.



Fonte: Elaboração própria.

5.2.7 r/WorldNews

A Tabela 10 contém as estatísticas encontradas para o grafo gerado a partir dos dados deste *subreddit*. É notável a quantidade exorbitante de número de nodos. Este é um fórum que frequentemente recebe destaque na página inicial do Reddit, provocando um fluxo muito alto de usuários, especialmente em postagens relacionadas a eventos muito impactantes. Naturalmente, os usuários responsáveis por essas postagens também recebem muitas réplicas, o que explica o valor de maior grau bem elevado. O grau médio, entretanto, foi relativamente baixo, indicando que muitos usuários interagiram pouco e provavelmente não são frequentadores desta Comunidade. Esse fenômeno também é observado na Figura 26, que mostra uma concentração grande de nodos nos intervalos mais baixos de grau.

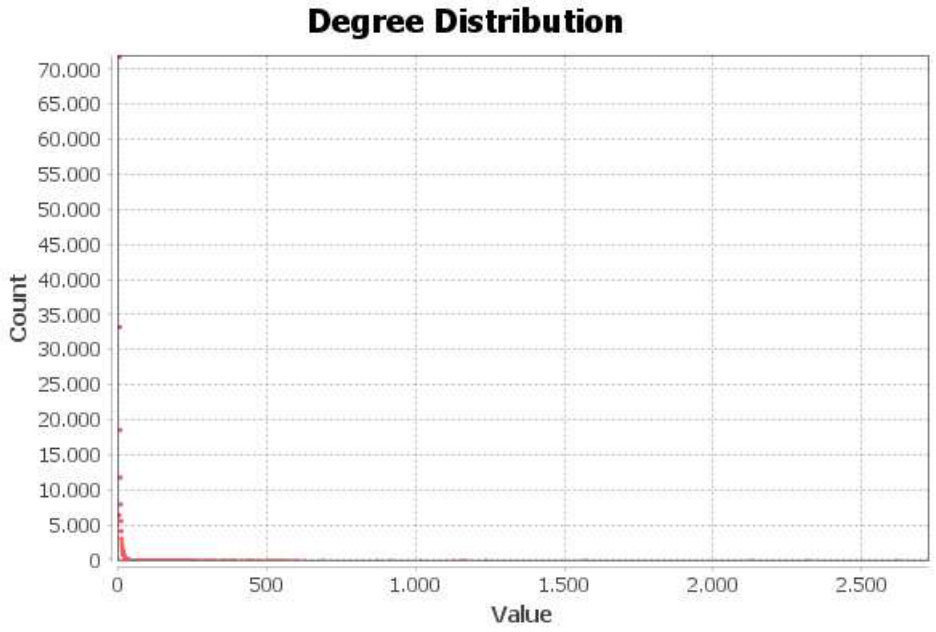
A Figura 27 revela o grafo gerado a partir dos dados deste *subreddit*. Quatro *hubs* ficaram destacados na imagem, possivelmente indicando usuários que realizaram postagens

Nº de Nós	Nº de Arestas	Maior Grau	Grau Médio	Diâmetro
183845	437223	2720	2, 37	20

Tabela 10 – Tabela de dados do grafo dirigido de r/WorldNews

Fonte: Elaboração própria.

Figura 26 – Distribuição do grau dos nodos do grafo de r/WorldNews.



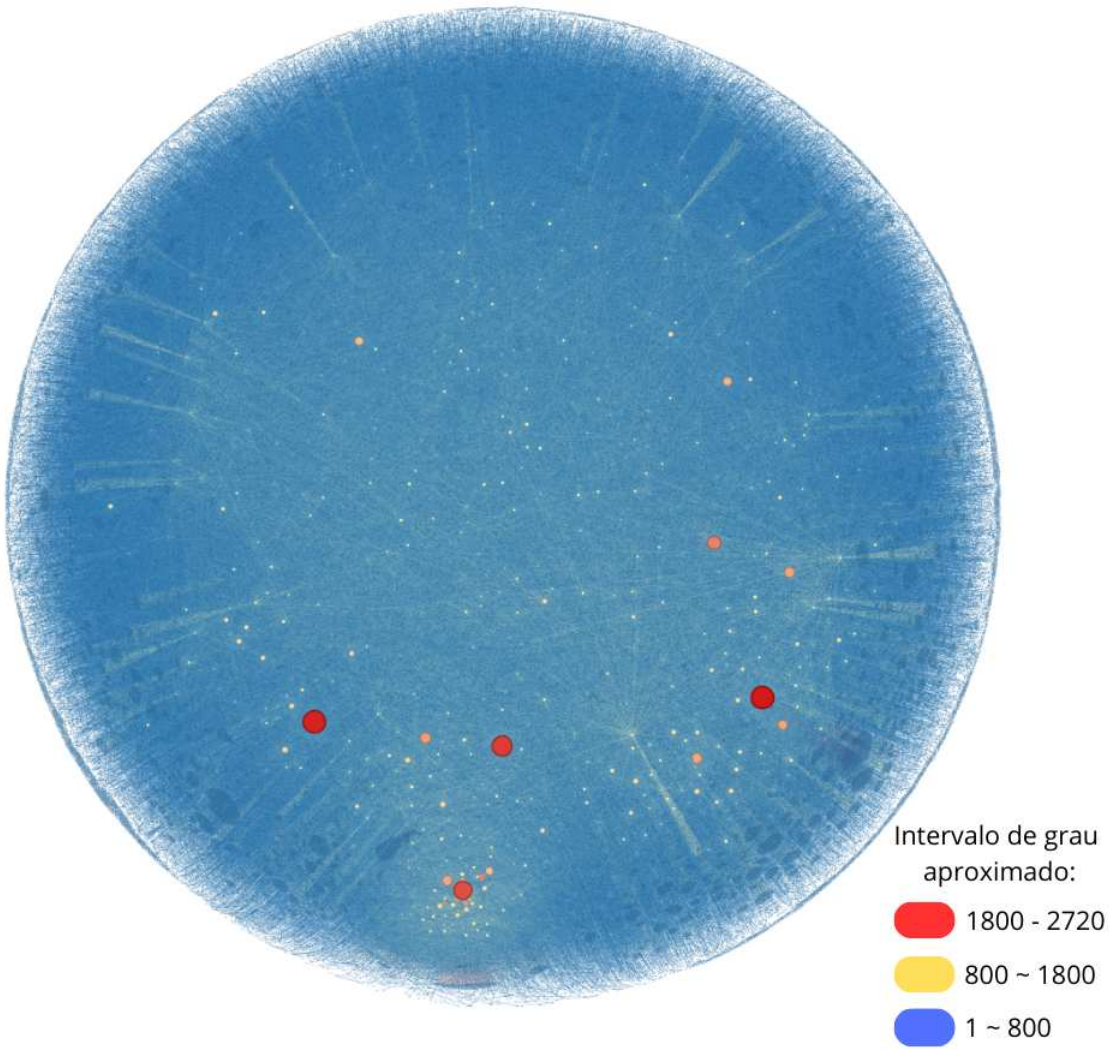
Fonte: Elaboração própria.

em momentos muito impactantes do conflito e por isso receberam uma quantidade imensa de réplicas. Esse *hubs* também ficaram relativamente próximos um do outro, indicando que esses usuários muito provavelmente realizaram suas postagens ou comentários em um mesmo período de tempo, onde a guerra Russo-Ucraniana estava atraindo muita atenção em escala global.

Por outro lado, a metade superior desse grafo está preenchida por nodos com grau baixo. Esses nodos podem representar usuários que não frequentam o *subreddit*, ou que interagiram apenas naquelas postagens que foram destacadas na página inicial do Reddit.

Os nodos amarelados que ocasionalmente aparecem na região mais central do grafo são mais indicativas do "dia a dia" dentro da Comunidade, isto é, dos períodos de tempo onde nenhum evento particularmente impactante aconteceu, e as postagens geralmente tem uma repercussão mais ordinária e pontual.

Figura 27 – Grafo dirigido dos dados coletados de r/WorldNews.



Fonte: Elaboração própria.

5.3 DETECÇÃO DE COMUNIDADES

Esta seção apresenta os resultados da execução de algoritmos de detecção de Comunidades sobre os grafos gerados previamente. O intuito desta parte do trabalho é observar o comportamento de cada algoritmo e compará-los entre si, além de mostrar as Comunidades encontradas de forma visual, novamente utilizando o software Gephi.

5.3.1 Algoritmo de Louvain

O algoritmo de Louvain foi executado sobre os grafos dos *subreddits* sendo analisados neste trabalho, com exceção de "r/News", "r/Politcs" e "r/WorldNews". O motivo da exclusão desses *subreddits* da seção de detecção de Comunidades é devido ao fato de que, apesar dos dados terem sido submetidos a uma filtragem para selecionar conteúdo

relacionado à guerra Russo-Ucraniana, esses *subreddits* tem propósitos menos direcionados a esse conflito em específico, e muitas discussões que mencionam a guerra não necessariamente têm foco nisso. Assim, durante a detecção de Comunidades, é possível que surjam Comunidades sem tanta relevância com o tópico principal deste trabalho.

A execução desse algoritmo busca encontrar a melhor "partição" de um grafo, medida pelo seu valor de modularidade. Assim, a partição contém as diversas Comunidades que foram encontradas. Para fins de facilitar a análise neste trabalho, foram selecionadas as 10 maiores Comunidades encontradas em cada grafo (baseado em seu número de nodos) e geradas as visualizações destas Comunidades no Gephi.

A Tabela 11 mostra o valor de modularidade e a quantidade de Comunidades encontradas após a execução do algoritmo de Louvain sobre os grafos de cada *subreddit*.

Subreddit	Modularidade	Nº Comunidades encontradas
r/EndlessWar	0.330	19
r/RussiaUkraineWar2022	0.346	71
r/Ukraine	0.345	195
r/UkraineWarVideoReport	0.393	158

Tabela 11 – Atributos encontrados pelo algoritmo de Louvain para cada *subreddit*.

Quanto mais próximo de 1 é o valor de modularidade, melhor é a estrutura de Comunidades detectada em cada um dos grafos. É perceptível que o algoritmo detectou valores no intervalo aproximado de 0.3 até 0.5, um resultado relativamente "positivo" para as estruturas encontradas. A seguir, serão apresentadas estatísticas e análises para cada uma das estruturas de Comunidade detectadas pelo algoritmo de Louvain.

5.3.1.1 r/EndlessWar

A Tabela 12 mostra os dados das 10 maiores Comunidades encontradas para o *subreddit*. Em relação a linha "todas" da Tabela, que é a combinação de todas as outras Comunidades juntas em um só grafo, é importante ressaltar que a quantidade de arestas, o grau médio e o grau máximo tem valores destoantes das Comunidades isoladas. Isso ocorre pois, ao juntar as Comunidades de 1 a 10 em uma única grande Comunidade, os usuários de Comunidades diferentes ainda possuem interações entre si, gerando novas arestas e aumentado seu grau.

Algumas informações interessantes podem ser extraídas da Tabela, como o fato de algumas Comunidades menores, como a 8, terem a média de grau não tão distante de Comunidades maiores, como a 5 e a 2. Possivelmente, isso ocorre quando usuários presentes nessa Comunidade tiveram discussões mais extensas entre si, gerando um grau elevado mesmo que a quantidade de participantes na discussão seja menor.

Além disso, vemos que a Comunidade 1 possui um grau máximo bem menor que a Comunidade 2, mas ainda obteve uma média muito maior, indicando uma concentração

Comunidade	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	292	476	3,25	78	7
2	262	390	2,97	165	6
3	205	262	2,55	123	9
4	203	294	2,89	51	8
5	197	278	2,82	103	8
6	185	232	2,5	38	9
7	165	226	2,73	47	6
8	160	225	2,81	55	7
9	151	192	2,54	32	8
10	147	166	2,25	90	6
Todas	1967	6102	6,2	531	Não calculado

Tabela 12 – Dados das Comunidades encontradas para r/EndlessWar.

maior de usuários com grau alto. Considerando a maneira que o Reddit é estruturado, é possível que usuários que interagiram com outros usuários que possuem grau alto de interações também tenham uma quantidade de interações elevada. Isso ocorre pela fato de postagens e comentários que tem grande quantidade de votos positivos no fórum ficarem em destaque na página, o que implica que as réplicas também ficarão destacadas junto, e terão uma visibilidade maior.

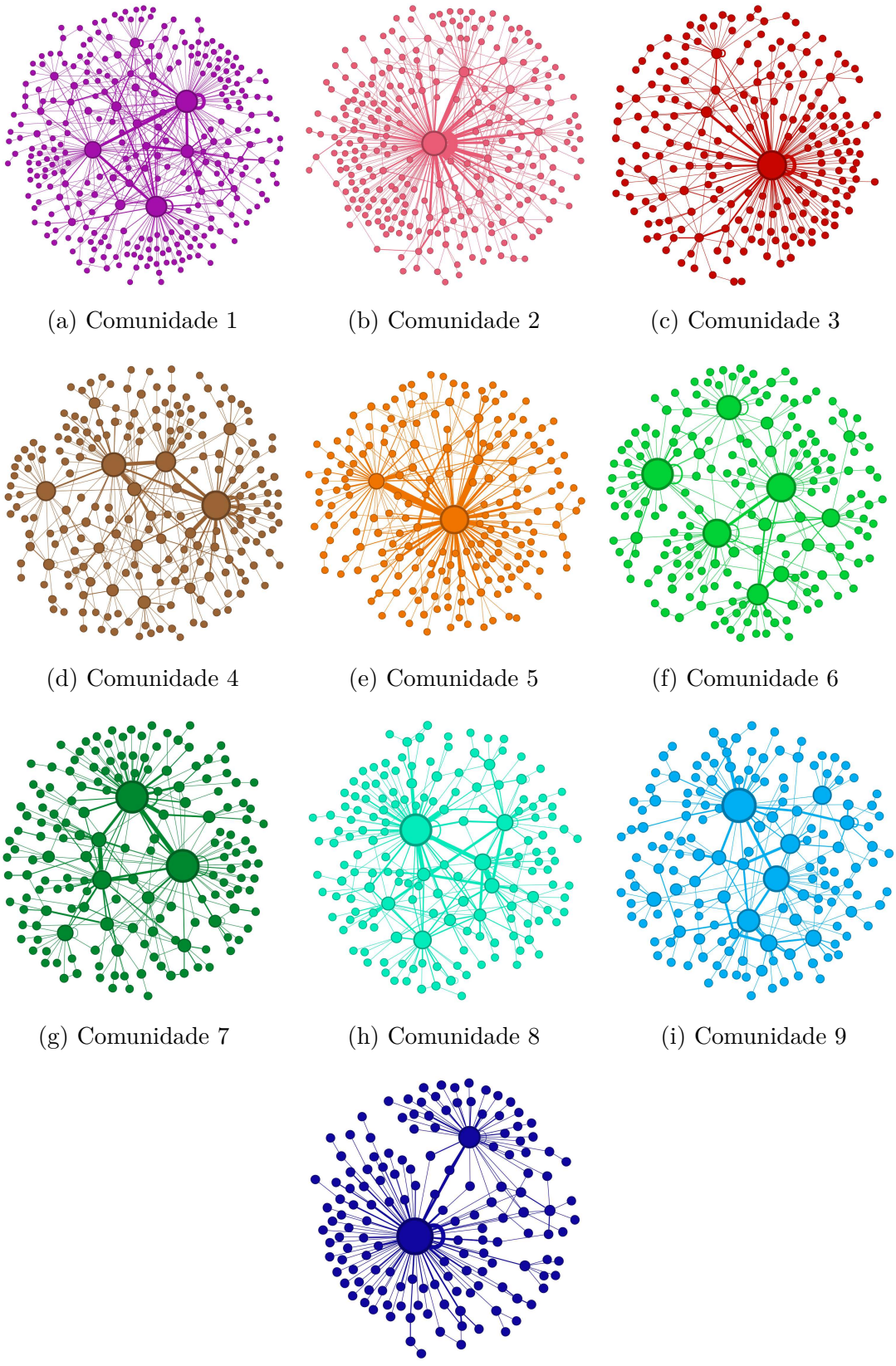
O diâmetro das Comunidades variou entre 6 e 9, o que pode ser indicativo de que as discussões ocorridas dentro destas Comunidades alcançou uma profundidade de até nove usuários diferentes participando de um mesmo tópico de discussão.

A Figura 28 mostra os grafos gerados a partir de cada uma das Comunidades. Essa visualização proporciona uma compreensão melhor da estrutura de cada Comunidade. É notável que em alguns casos, como ocorre nas Comunidades 2, 3, 5 e 10, o algoritmo parece ter detectado um agrupamento centrado em apenas um ou dois usuários *hubs*, possivelmente indicando discussões de grande repercussão provocadas por estes usuários.

Outras Comunidades, como a de números 1, 4, 6, 7, 8 e 9 possuem *hubs* menos destoantes do resto do grafo, um sinal de que a Comunidade não tão influenciada pela repercussão de alguns poucos usuários, mas provavelmente por uma variedade maior de pessoas e assuntos que apareceram de forma frequente dentro do *subreddit*.

A Figura 29 representa a visualização da junção dos grafos apresentados na Figura 28. As Comunidades estão coloridas da mesma forma nas duas Figuras. É perceptível que os *hubs* de cada Comunidade estão próximos um do outro, na região central do grafo. Um fenômeno interessante é que alguns nodos da mesma Comunidade ficam mais distantes um do outro no grafo conjunto que outros nodos de Comunidades diferentes. O algoritmo de Louvain tenta maximizar a modularidade, e por causa disso é possível que nodos que estejam fortemente conectados ainda podem ser separados em Comunidades diferentes.

Figura 28 – Visualização dos grafos das Comunidades de r/EndlessWar.



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

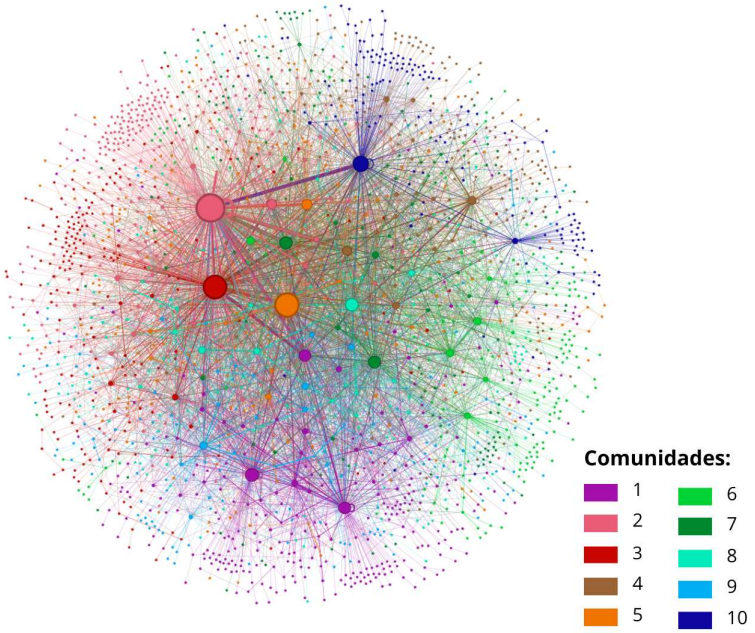
(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

(g) Comunidade 7 (h) Comunidade 8 (i) Comunidade 9

(j) Comunidade 10

Fonte: Elaboração própria.

Figura 29 – Grafo da junção das Comunidades de r/EndlessWar, detectadas pelo algoritmo de Louvain.



Fonte: Elaboração própria.

A fim de compreender melhor os tópicos de discussão que acontecem em cada Comunidade, a Figura 30 apresenta nuvens de palavras geradas a partir das 3 maiores Comunidades deste *subreddit*. As palavras “China”, “Military” (militar) e “American” (americano) aparecem nas três Comunidades. Uma possível causa é o caráter norte-americano da rede social como um todo, juntamente com as tensões entre Estados Unidos e China, que se elevaram em alguns momentos do conflito Russo-Ucraniano.

Figura 30 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Louvain em r/EndlessWar.

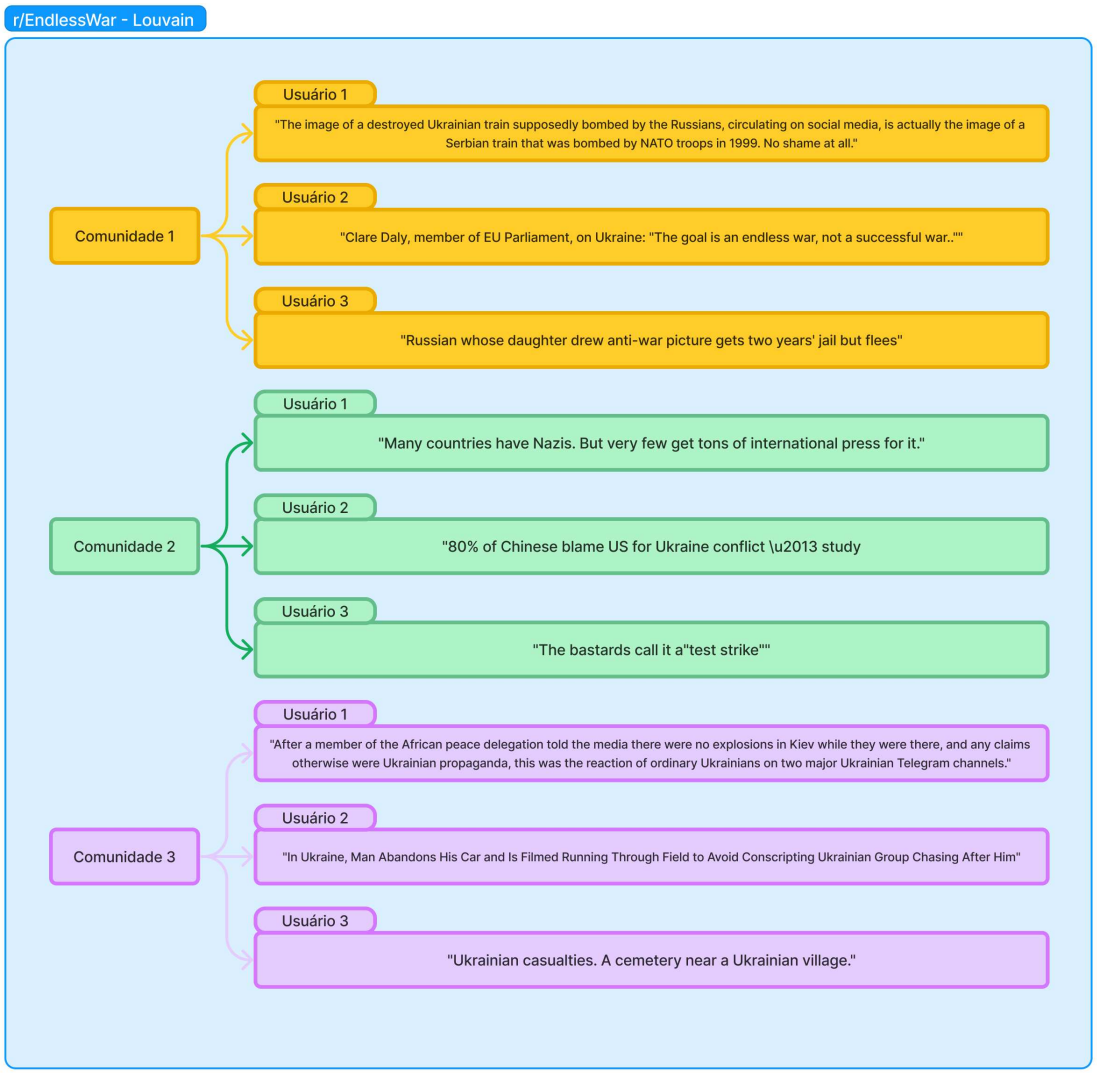
(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

A Figura 31 mostra alguns dos comentários com mais repercussão (pontuação dentro do Reddit), vindos dos usuários de maior grau das três primeiras Comunidades Observando os comentários dos usuários, é perceptível que muitos fazem menção a tragédia da guerra. Na Comunidade 2, também é mostrado um comentário de caráter político,

fazendo referência a relação entre EUA e China.

Figura 31 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/EndlessWar.



Fonte: Elaboração própria.

5.3.1.2 r/RussiaUkraineWar2022

A Tabela 13 mostra as estatísticas para as 10 maiores Comunidades encontradas pelo algoritmo de Louvain. O grau médio teve uma diferença considerável entre a maior e a menor Comunidade. Além disso, o grau máximo não foi necessariamente proporcional ao tamanho da Comunidade, diferentemente de grafos anteriores. O diâmetro, por sua vez, foi relativamente parecido entre todas as Comunidades.

Em relação aos grafos de cada Comunidade, observados na Figura 32, os comportamentos observados são similares às Comunidades analisadas previamente. Vemos

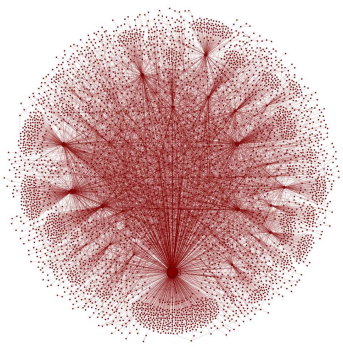
Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	4194	7683	3,66	971	9
2	3117	4934	3,16	1194	9
3	2507	3749	2,99	835	10
4	1850	2566	2,77	408	11
5	1809	2573	2,84	360	10
6	1731	2556	2,95	234	12
7	1484	2019	2,72	542	11
8	1467	1937	2,64	408	12
9	1266	1679	2,65	318	11
10	1077	1390	2,58	406	11
Todas	20502	62057	6,05	3209	Não calculado

Tabela 13 – Dados das Comunidades encontradas para r/RussiaUkraineWar2022.

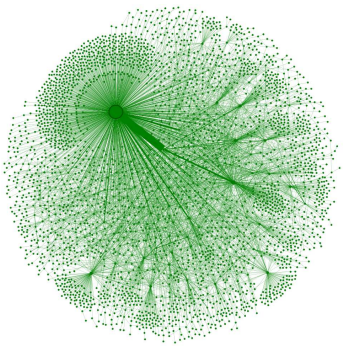
Comunidades com uma grande quantidade de *hubs*, como a 1 e 3, juntamente com Comunidades de poucos *hubs* como a 6, 7 e 8, onde apenas alguns poucos nodos tem tamanho de destaque devido ao seu grau.

O grafo conjunto, na Figura 33 teve três regiões de destaque, correspondentes as Comunidades 1, 2 e 3, que dividem a rede em terços de tamanho parecido. Parece existir uma tendência, não apenas neste grafo como nos anteriores, de que as Comunidades de maior tamanho tendem a ficar relativamente distantes no grafo conjunto, implicando que não existe tanta conexão entre elas. É possível que cada uma dessas Comunidades esteja relacionada a uma postagem ou um conjunto de postagens que foram feitos em períodos de tempo diferentes dentro do *subreddit*, de forma que os usuários presentes em cada Comunidade foram frequentadores de "r/RussiaUkraineWar2022" em momentos diferentes do conflito, o que explica a baixa conectividade entre certas Comunidades.

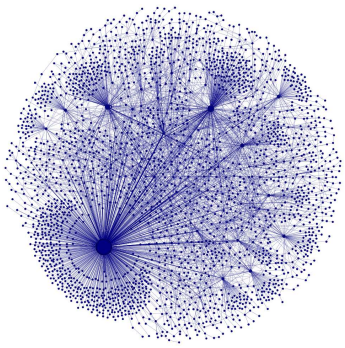
Figura 32 – Visualização dos grafos das Comunidades de r/RussiaUkraineWar2022.



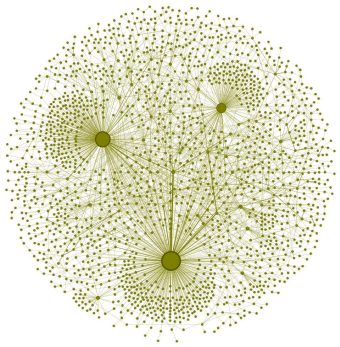
(a) Comunidade 1



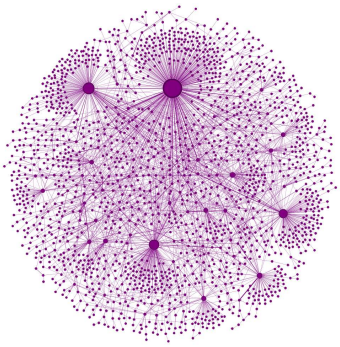
(b) Comunidade 2



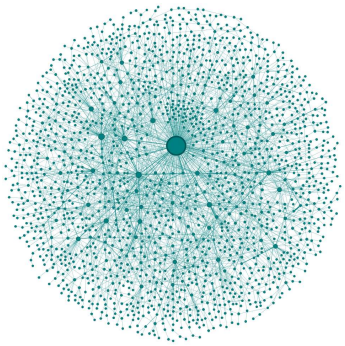
(c) Comunidade 3



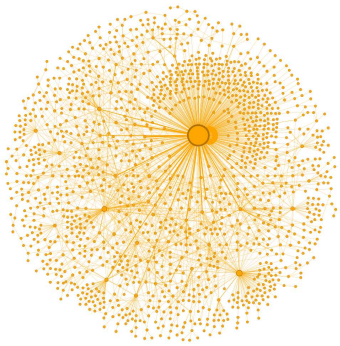
(d) Comunidade 4



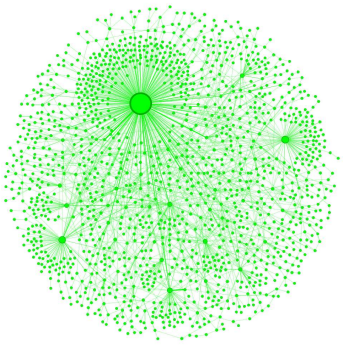
(e) Comunidade 5



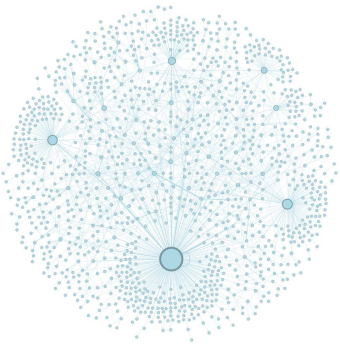
(f) Comunidade 6



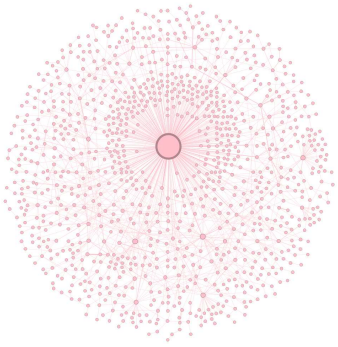
(g) Comunidade 7



(h) Comunidade 8



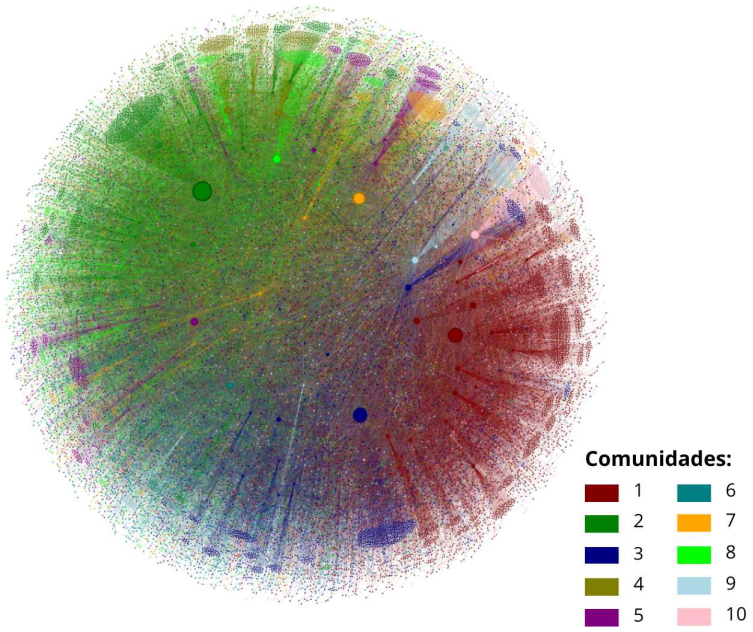
(i) Comunidade 9



(j) Comunidade 10

Fonte: Elaboração própria.

Figura 33 – Grafo da junção das Comunidades de r/RussiaUkraineWar2022, detectadas pelo algoritmo de Louvain.



Fonte: Elaboração própria.

A Figura 34 mostra as nuvens de palavras para as três maiores Comunidades detectadas. Estão presentes diversos termos militares, como “soldier” (soldado), “tank” (tanque) e “combat” (combate). Além disso, aparecem algumas palavras com conotação positiva, como “hope” (esperança), “love” (amor) e “life” (vida). A palavra “Slava” vem novamente do termo “Slava Ukraini” (glória a Ucrânia), indicando que essas Comunidades provavelmente tiveram muitas postagens de apoio ao povo ucraniano.

Figura 34 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Louvain em r/RussiaUkraineWar2022.

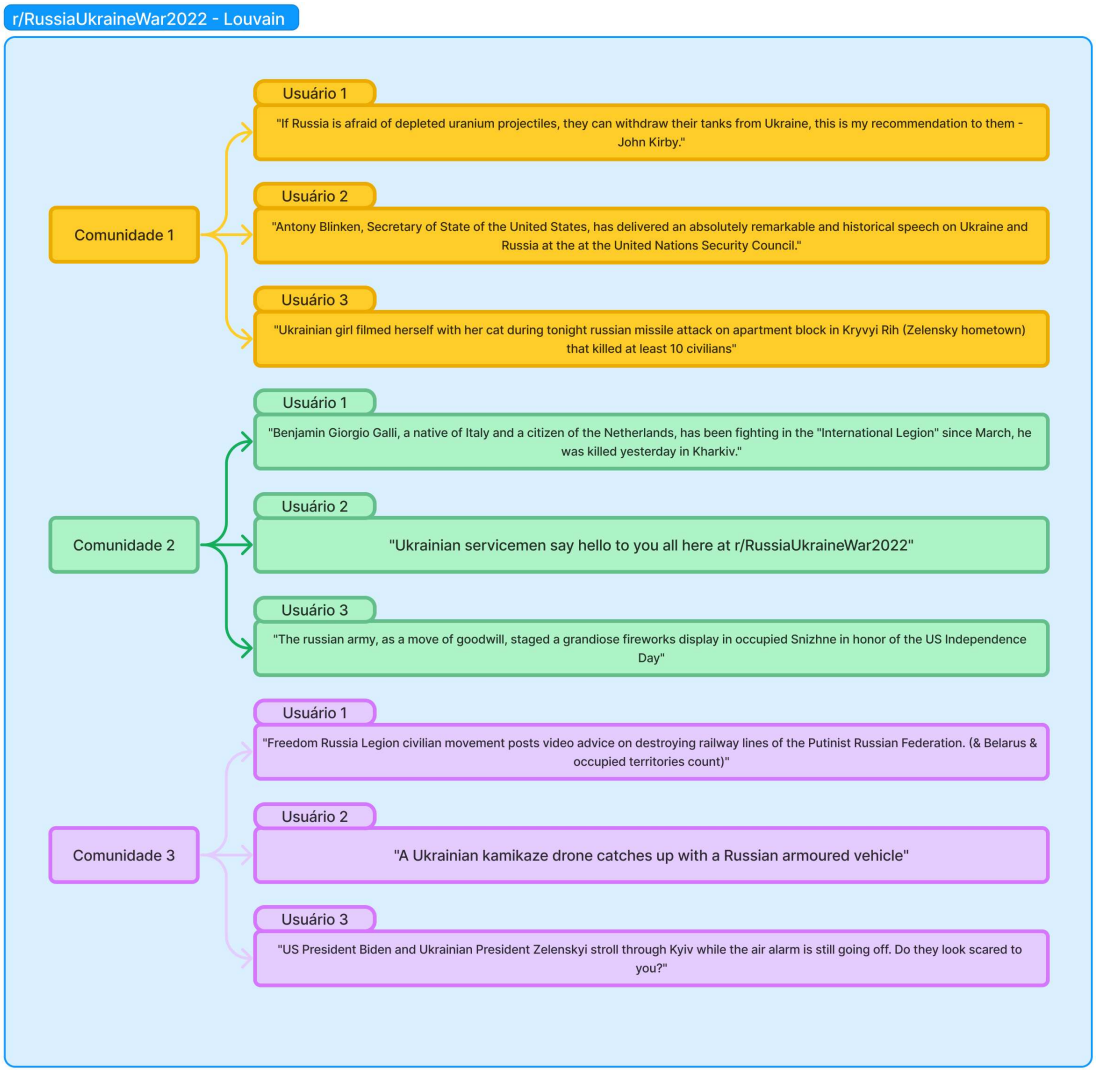
(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

A Figura 35 mostra os comentários ou postagens que atingiram mais repercussão dos usuários de grau mais elevado nas três primeiras Comunidades detectadas. Analisando a imagem, parece que comentários referenciando os EUA atingiram uma repercussão alta. Além disso, os comentários parecem trazer um ponto de vista pro-Ucraniano, onde os ataques da Rússia são vistos como chocantes e traumáticos (terceiro comentário da Comu-

nidade 1), enquanto a investida Ucraniana geralmente recebe repercussão com conotação mais positiva (segundo comentário da Comunidade 3). O terceiro comentário da Comunidade 2 ironiza o ataque Russo, comparando com "fogos de artifício em homenagem ao Dia da Independência".

Figura 35 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/RussiaUkraineWar2022.



Fonte: Elaboração própria.

5.3.1.3 r/Ukraine

A Tabela 14 mostra as estatísticas encontradas para as 10 maiores Comunidades detectadas pelo algoritmo de Louvain neste *subreddit*. Dessa vez, a quantidade de nodos é extremamente diferente da primeira até a última Comunidade, e a proporção do número de arestas não foi necessariamente proporcional ao número de nodos. O grau médio também não foi proporcional, como é possível notar observando as Comunidades 2 e 8, por exemplo.

O grau máximo também foi extremamente variável, de forma que até mesmo Comunidades menores como a 10 obtiveram um valor comparativamente alto. O diâmetro de todas as Comunidades se manteve bem parecido.

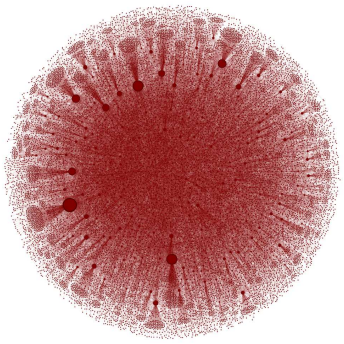
Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	16772	31092	3,7	692	13
2	11327	23280	4,11	1505	11
3	7512	11628	3,09	760	13
4	6651	9289	2,79	1181	15
5	6346	10567	3,33	296	13
6	6310	10399	3,29	283	13
7	6195	8798	2,84	847	14
8	5805	8992	3,09	349	13
9	4786	6679	2,79	683	13
10	4213	5288	2,51	911	14
Todas	75917	256998	6,77	3548	Não calculado

Tabela 14 – Dados das Comunidades encontradas para r/Ukraine.

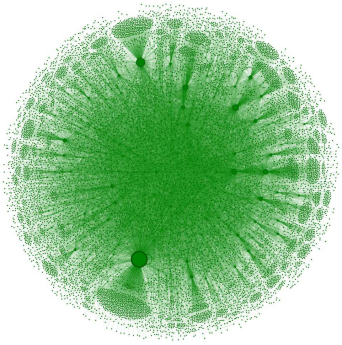
Observando os grafos de cada Comunidade na Figura 36, é notável que no geral cada Comunidade teve vários *hubs*, ao invés de apenas um só. Isso significa que o algoritmo não detectou Comunidades que parecem ser centradas em apenas um usuário com grande repercussão, mas sim em diversos usuários diferentes que participaram ativamente de determinada discussão e obtiveram um grande número de interações.

O grafo conjunto, observado na Figura 37, também teve um comportamento um pouco diferente dos outros apresentados nas análises dos *subreddits* anteriores. Dessa vez, a Comunidade 1 ficou destacada na parte de baixo do grafo, de forma "isolada", possivelmente indicando que possui uma proximidade maior apenas com a Comunidade 2, logo acima dela. A Comunidade 2 tomou a parte central do grafo, e devido a isso pode estar mais interligada com outras Comunidades que foram detectadas. A parte superior do grafo conjunto foi preenchida com as outras Comunidades, como a 3, 4 e 7, que parecem estar muito interconectadas entre si.

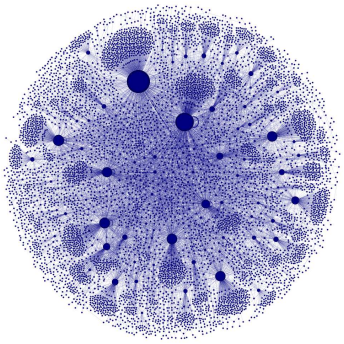
Figura 36 – Visualização dos grafos das Comunidades de r/Ukraine.



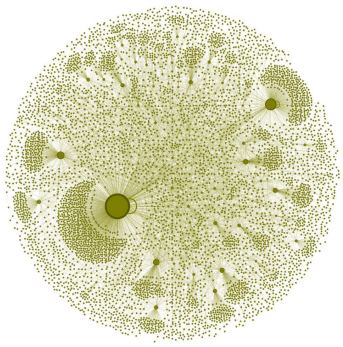
(a) Comunidade 1



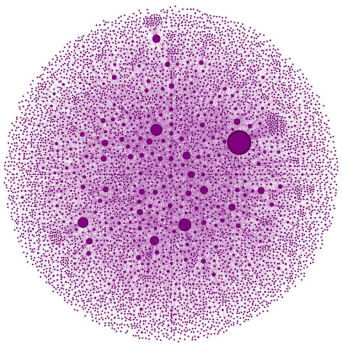
(b) Comunidade 2



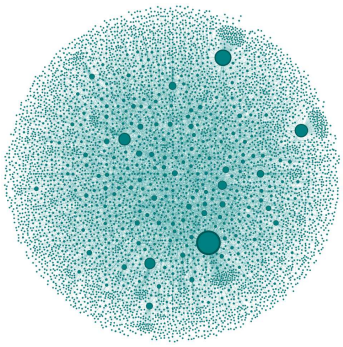
(c) Comunidade 3



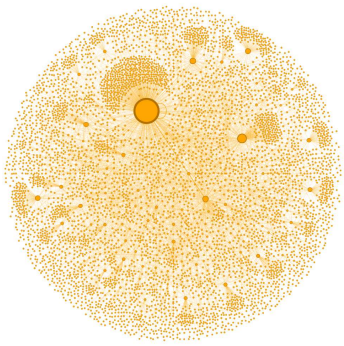
(d) Comunidade 4



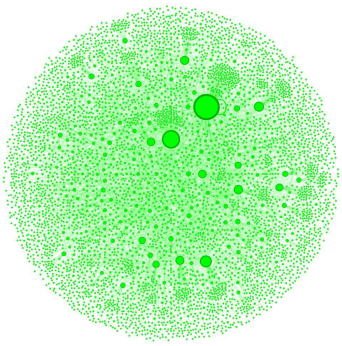
(e) Comunidade 5



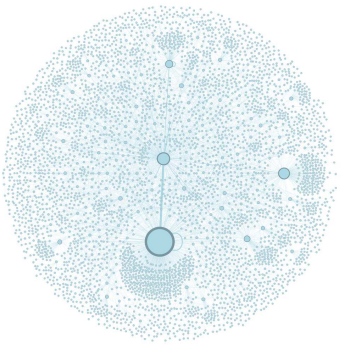
(f) Comunidade 6



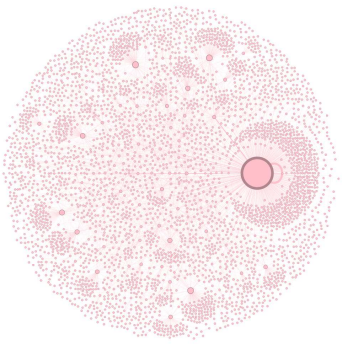
(g) Comunidade 7



(h) Comunidade 8



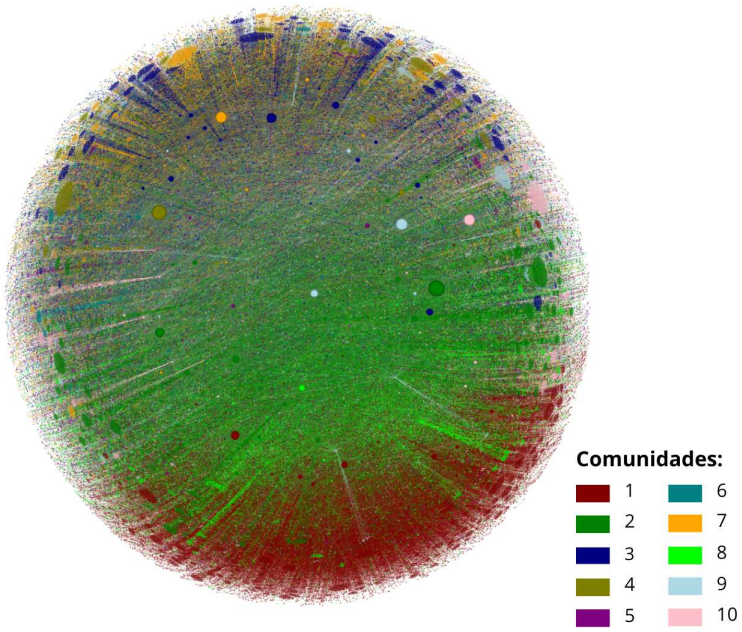
(i) Comunidade 9



(j) Comunidade 10

Fonte: Elaboração própria.

Figura 37 – Grafo da junção das Comunidades de r/Ukraine, detectadas pelo algoritmo de Louvain.

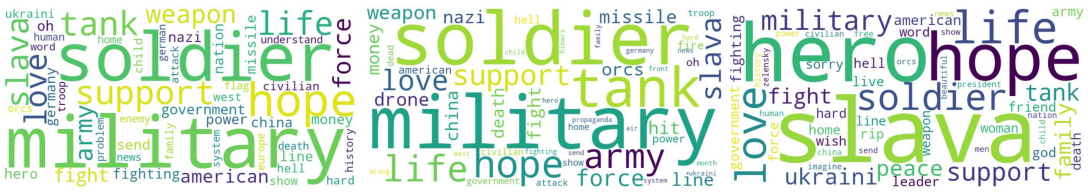


Fonte: Elaboração própria.

A Figura 38 mostra as nuvens de palavra geradas a partir dos dados das três maiores Comunidades detectadas por Louvain em "r/Ukraine". As três Comunidades parecem ser extremamente similares em relação as palavras mais utilizadas. No geral, parece existir uma predominância por termos militares, como "missiles" (mísseis), "soldier" (soldado) e "tank" (tanque). A Comunidade 3, particularmente, se destacou pela predominância de palavras positivas, como "hero" (herói), "hope" (esperança) e "life" (vida), provavelmente expressando apoio ao povo e ao exército ucraniano.

A Figura 39 mostra os comentários ou postagens de maior repercussão nas três maiores Comunidades detectadas. Muitas das mensagens fazem referência a reconquista de território ucraniano, como a retomada da Ilha da Serpente e a cidade de Kherson, mostrando que esse tipo de conteúdo foi bastante compartilhado e discutido dentro desse *subreddit*.

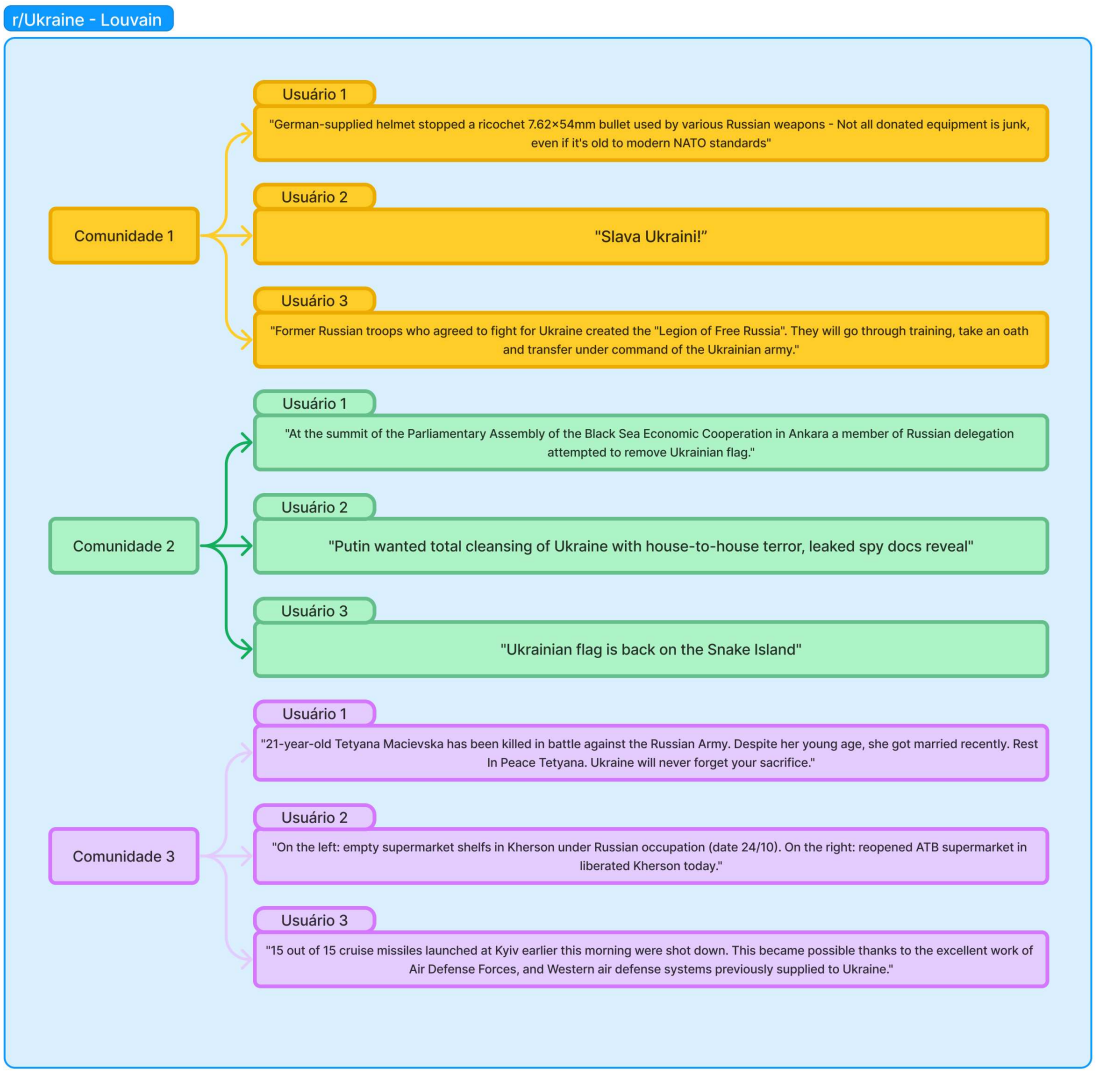
Figura 38 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Louvain em r/Ukraine.



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

Figura 39 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/Ukraine.



Fonte: Elaboração própria.

5.3.1.4 r/UkraineWarVideoReport

A Tabela 15 mostra as estatísticas para as 10 maiores Comunidades encontradas pelo algoritmo de Louvain para este *subreddit*. Existe grande variação entre o número de nodos e número de arestas da Comunidade 1 para a Comunidade 10. Além disso, o grau médio também difere bastante, e as Comunidades 2 e 3 tiveram grau médio maior do que a Comunidade 1, apesar de serem menores em quantidade de nodos. O grau máximo das Comunidades também variou bastante, com a Comunidade 3 tendo um valor muito acima, e as Comunidades 8 e 9 muito abaixo do restante das Comunidades. O diâmetro também foi bem diferente entre cada Comunidade, com destaque para as Comunidades 2 e 3, que tiveram o menor valor de diâmetro apesar de seu tamanho e de seu grau médio elevado.

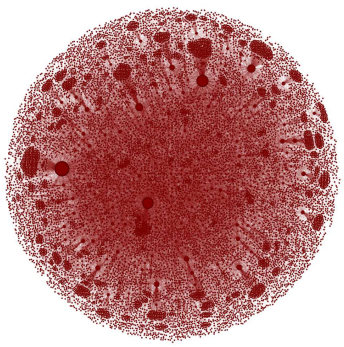
Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	16645	29284	3,51	837	13
2	14650	31247	4,26	1972	10
3	11921	24039	4,03	2743	10
4	8888	14923	3,35	590	12
5	6463	9811	3,03	900	13
6	3502	4578	2,61	480	14
7	2907	3686	2,53	602	14
8	1844	2200	2,38	86	17
9	1712	1979	2,31	65	18
10	1667	1932	2,31	425	17
Todas	70199	221473	6,31	5925	Não calculado

Tabela 15 – Dados das Comunidades encontradas para r/UkraineWarVideoReport.

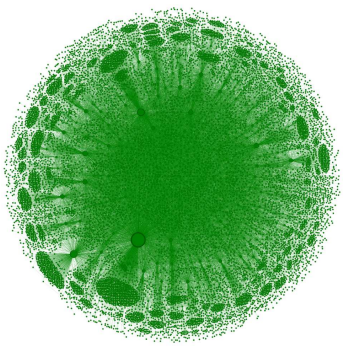
A visualização dos grafos desta Comunidade, presentes na Figura 40, mostram que de forma geral cada Comunidade teve diversos *hubs*, bem distribuídos através de toda a extensão de cada grafo. Isso indica que, dentro de cada Comunidade, diversos usuários diferentes obtiveram grande repercussão em suas postagens e comentários. As três maiores Comunidades, em particular, parecem ter uma estrutura extremamente semelhante: diversos *hubs* espalhados por todas as direções dos grafos, próximos às bordas.

O grafo que mostra a junção de todas essas Comunidades, presente na Figura 41, também expõe certa uniformidade e semelhança entre a estrutura das maiores Comunidades. Esse grafo ficou dividido em quatro grande regiões, correspondentes às quatro maiores Comunidades que foram detectadas. A região azul, representante da Comunidade 3, parece ter se estendido mais para o centro do grafo, possivelmente indicando uma conexão maior com todas as outras Comunidades.

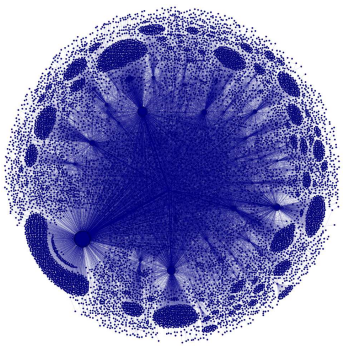
Figura 40 – Visualização dos grafos das Comunidades de r/UkraineWarVideoReport.



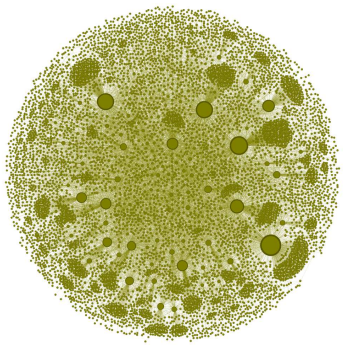
(a) Comunidade 1



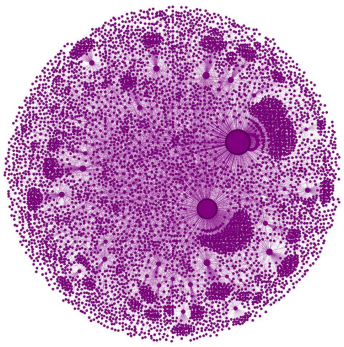
(b) Comunidade 2



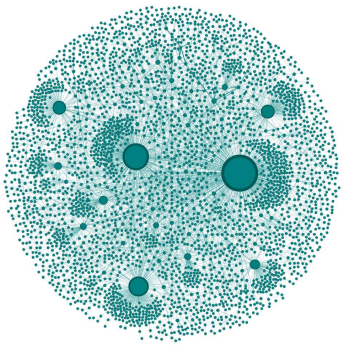
(c) Comunidade 3



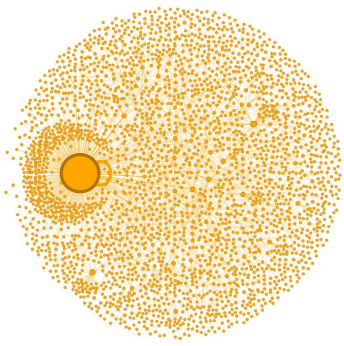
(d) Comunidade 4



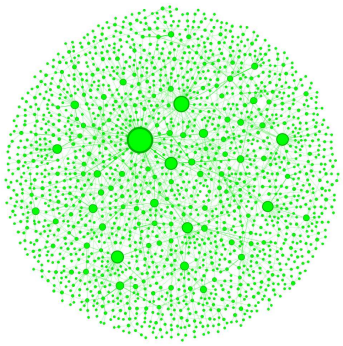
(e) Comunidade 5



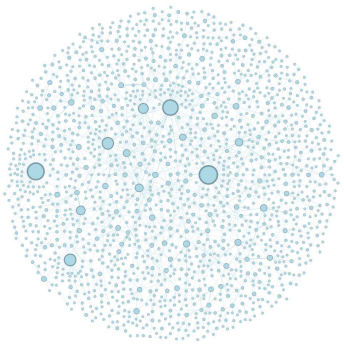
(f) Comunidade 6



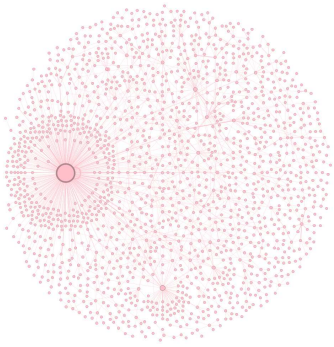
(g) Comunidade 7



(h) Comunidade 8



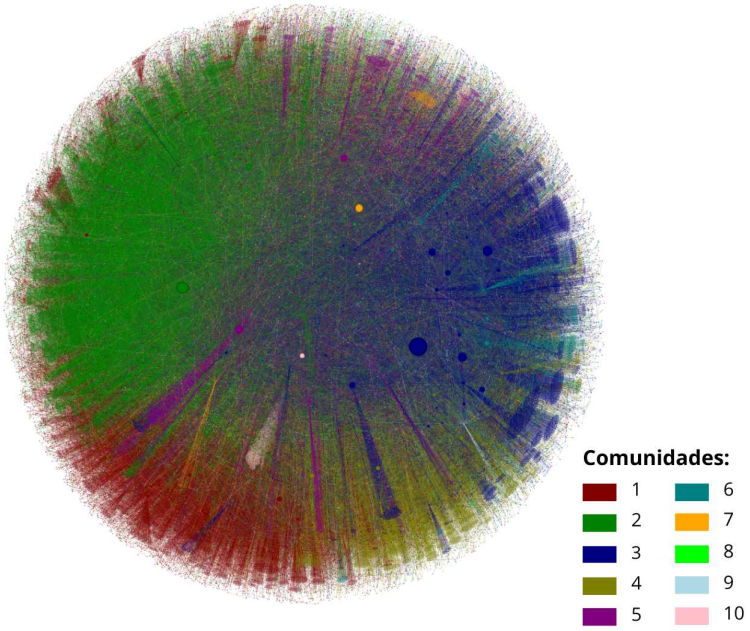
(i) Comunidade 9



(j) Comunidade 10

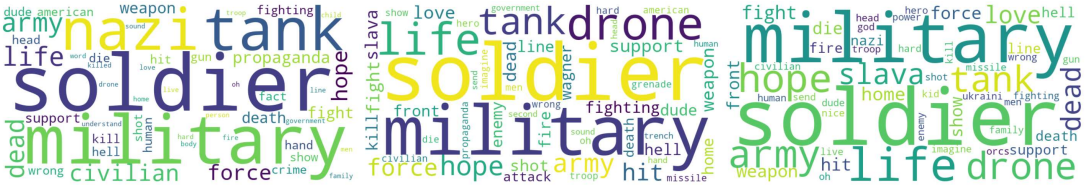
Fonte: Elaboração própria.

Figura 41 – Grafo da junção das Comunidades de r/UkraineWarVideoReport, detectadas pelo algoritmo de Louvain.



Fonte: Elaboração própria.

Figura 42 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Louvain em r/UkraineWarVideoReport.



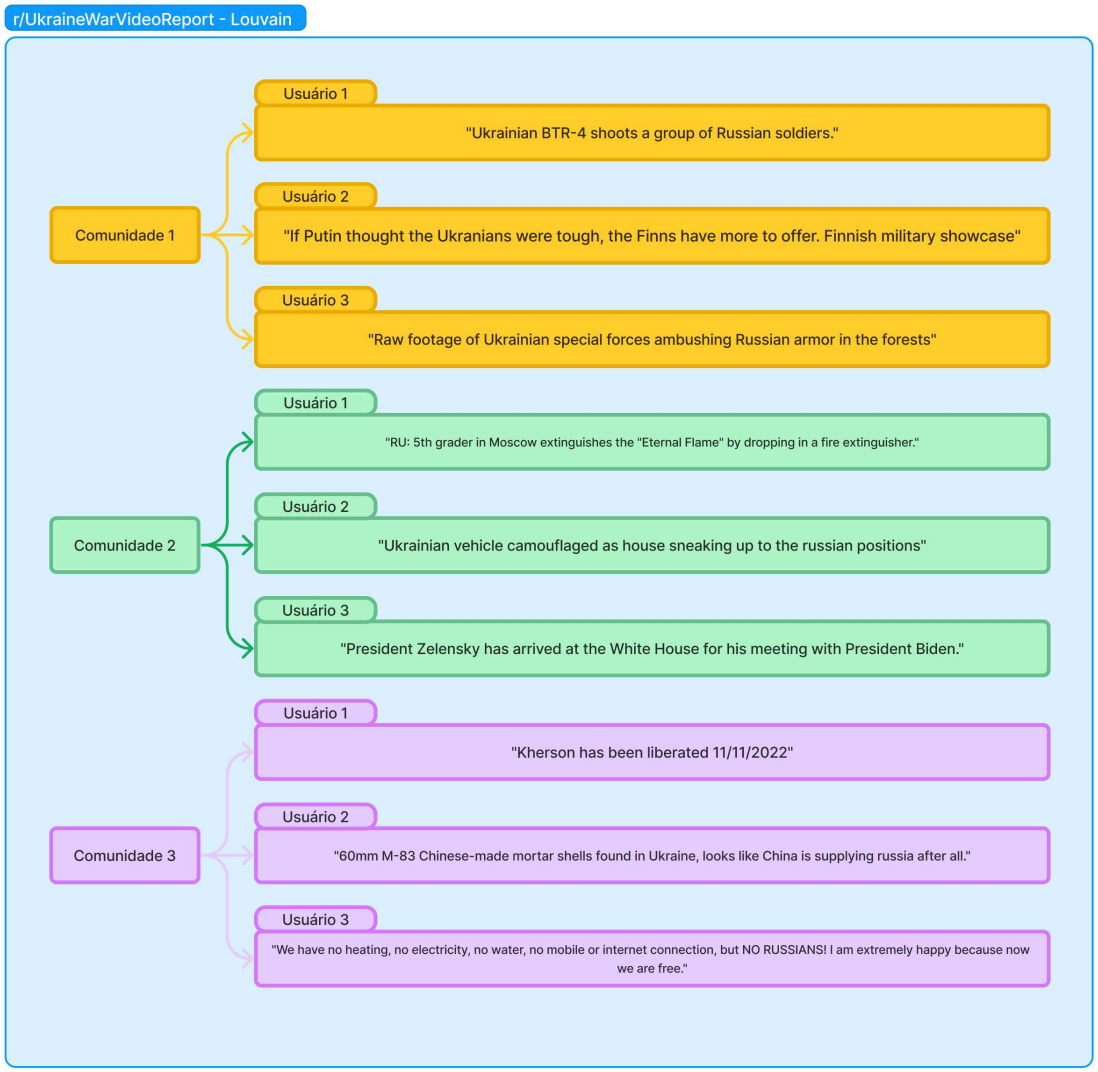
(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

A Figura 42 mostra as nuvens de palavras das três maiores Comunidades detectadas. Novamente, as três imagens são bem similares e destacam diversos termos militares. Diferentemente dos *subreddits* anteriores, aqui existem menos palavras positivas, provavelmente porque o fórum é dedicado ao compartilhamento de imagens e vídeos mais sensíveis.

A Figura 43 mostra comentários ou postagens de alta repercussão nessas três Comunidades. A Comunidade 1 parece mais focada em compartilhar conteúdo de caráter militar. A Comunidade 2, por sua vez, tem um foco mais político, mencionando os presidentes da Ucrânia e dos EUA, e ironizando a Chama Russa, monumento em homenagem a soldados soviéticos. A Comunidade 3 faz menção à retomada de território por parte da Ucrânia.

Figura 43 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/UkraineWarVideoReport.



Fonte: Elaboração própria.

5.3.2 Algoritmo de Leiden

Assim como o algoritmo de Louvain, o algoritmo de Leiden foi executado sobre as Comunidades *subreddits* analisadas neste trabalho. A partir do resultado, foram separadas as 10 maiores Comunidades de cada *subreddit*, que serão analisadas à seguir. A Tabela 16 mostra os valores de modularidade e o número de Comunidades detectadas pelo algoritmo de Leiden para cada *subreddit*. De forma geral, é possível observar que os valores de modularidade e número de Comunidades não foram muito diferentes em relação aos encontrados pelo algoritmo de Louvain, presentes na Tabela 11.

Subreddit	Modularidade	Nº Comunidades encontradas
r/EndlessWar	0.288	21
r/RussiaUkraineWar2022	0.335	73
r/Ukraine	0.362	225
r/UkraineWarVideoReport	0.401	151

Tabela 16 – Atributos encontrados pelo algoritmo de Leiden para cada *subreddit*.

5.3.2.1 r/EndlessWar

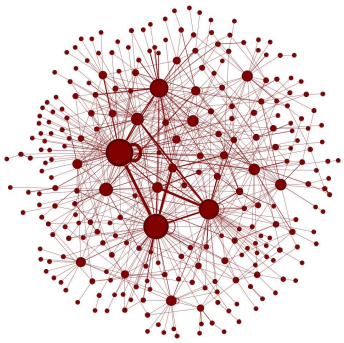
A Tabela 17 fornece as estatísticas das 10 maiores Comunidades detectadas pelo algoritmo de Leiden no *subreddit*. A Comunidade 1 possui um número de nodos relativamente elevado comparado as outras, que possuem valores num intervalo menor. O número de arestas também é bastante elevado na primeira Comunidade, mas com variação menor a partir da Comunidade 3 até a 10. As Comunidades 1, 2 e 6 tiveram valores mais elevados de grau médio, entretanto a Comunidade 7 se destacou pelo maior valor de grau máximo encontrado. Por fim, o diâmetro é similar entre todas as Comunidades.

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	262	556	4,24	45	6
2	196	379	3,66	34	7
3	180	209	2,32	31	6
4	171	202	2,36	87	7
5	170	219	2,57	36	8
6	165	299	3,62	20	6
7	163	193	2,36	126	7
8	152	229	3	12	9
9	145	229	3,15	27	7
10	140	239	3,41	18	7
Todas	1744	5193	5,95	531	Não calculado

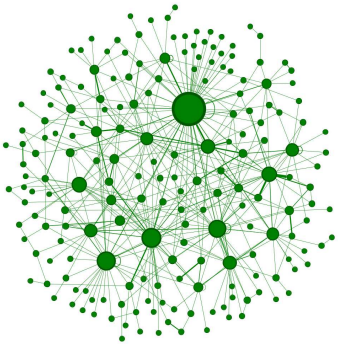
Tabela 17 – Dados das Comunidades encontradas para r/EndlessWar.

Na Figura 44 vemos a visualização dos grafos detectados por Leiden neste *subreddit*. De forma geral, os grafos não ficaram centradas em apenas um usuários, mas sim em vários pontos com grau mais elevado, como é perceptível nas Comunidades 1, 2 e 6, por exemplo. No grafo conjunto, mostrado na Figura 45, vemos que as Comunidades maiores ficaram bem próximas umas das outras, uma indicação de que estão fortemente conectadas. De maneira geral, nenhum Comunidade parece isolada das outras, uma possível consequência do tamanho menor deste *subreddit* comparado aos outros analisados neste trabalho.

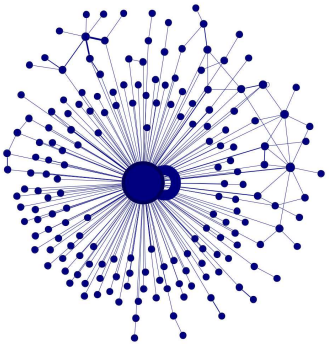
Figura 44 – Visualização dos grafos das Comunidades de r/EndlessWar.



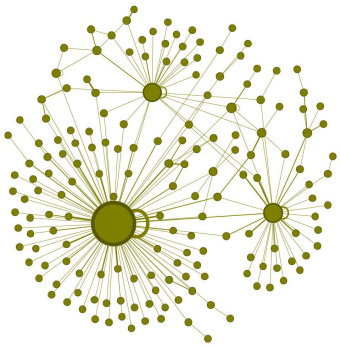
(a) Comunidade 1



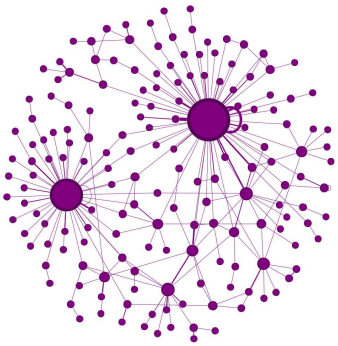
(b) Comunidade 2



(c) Comunidade 3

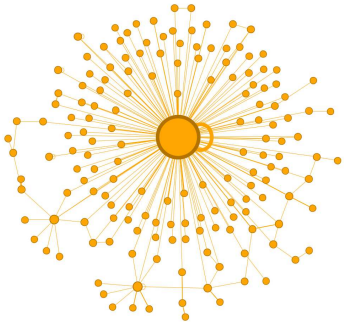


(d) Comunidade 4

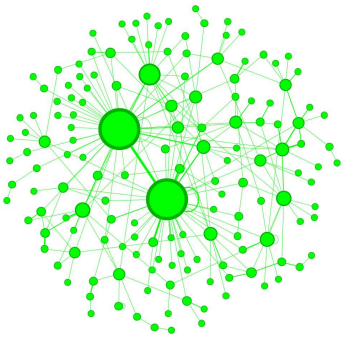


(e) Comunidade 5

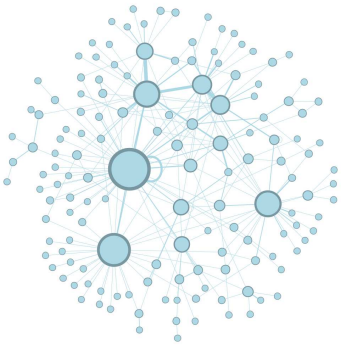
(f) Comunidade 6



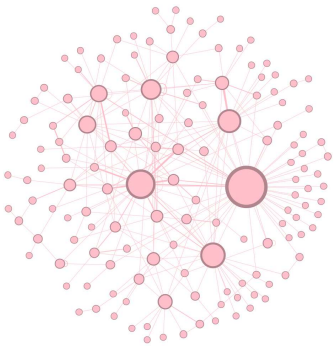
(g) Comunidade 7



(h) Comunidade 8



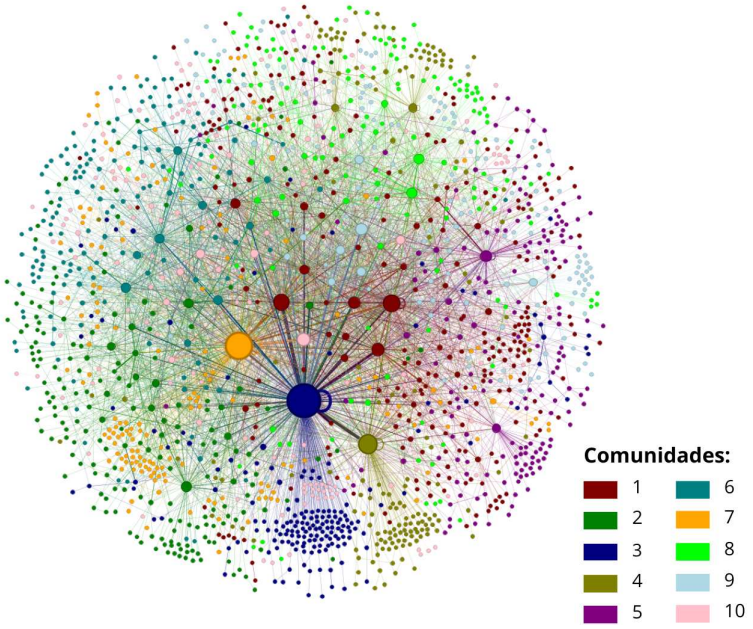
(i) Comunidade 9



(j) Comunidade 10

Fonte: Elaboração própria.

Figura 45 – Grafo da junção das Comunidades de r/EndlessWar, detectadas pelo algoritmo de Leiden.



Fonte: Elaboração própria.

A Figura 46 mostra as nuvens de palavras geradas a partir das três maiores Comunidades detectadas. Comparado às nuvens de palavras analisadas previamente na seção 5.3.1, estas não possuem a presença de palavras com conotação positiva. Aqui, termos militares e políticos foram mais proeminentes em todas as Comunidades.

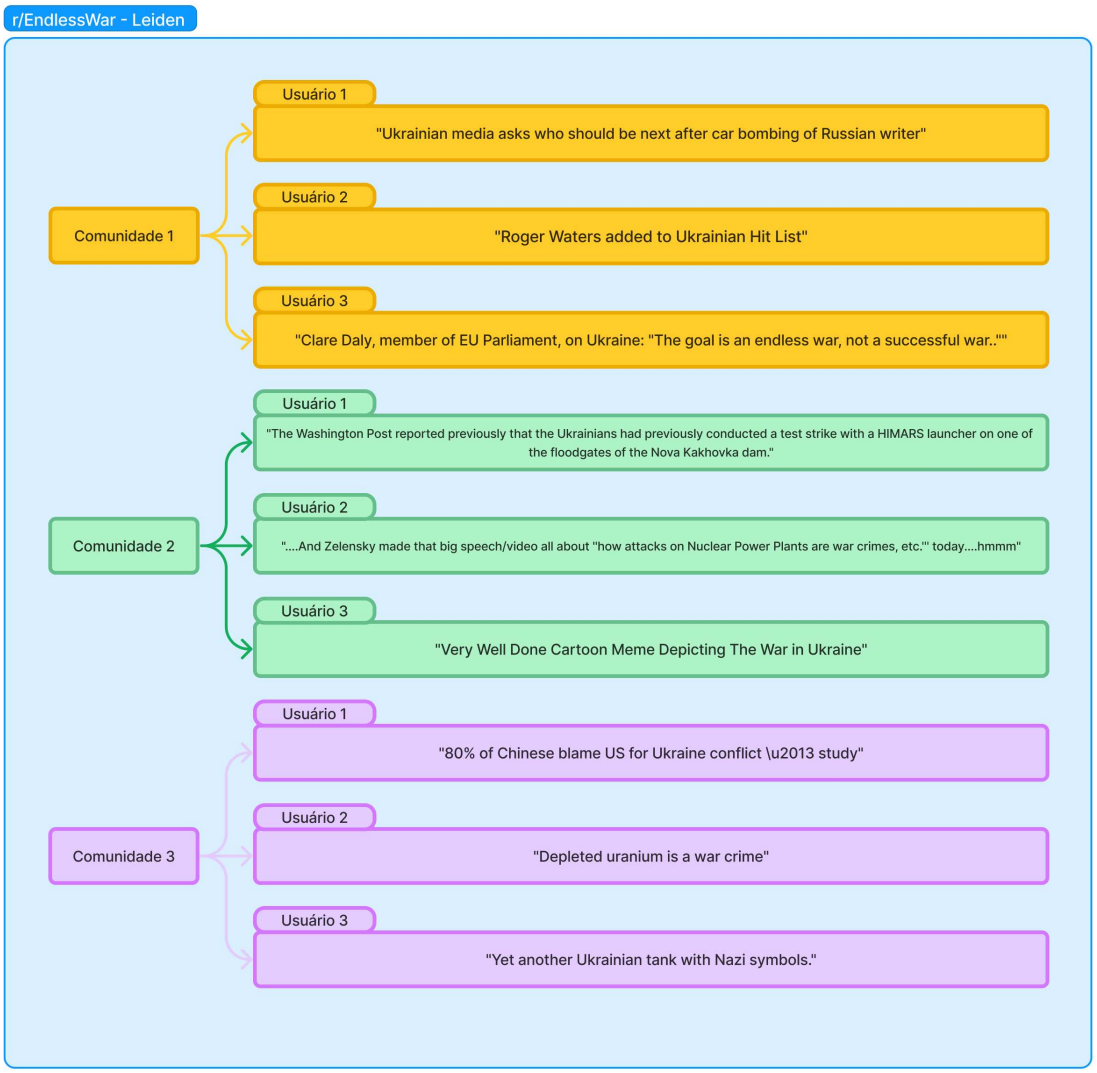
A palavra "Nazi" ficou em evidência, e o contexto de seu uso pode ser melhor entendido observando a Figura 47. Muitos dos comentários expostos aqui revelam uma visão crítica direcionada à Ucrânia. Na Comunidade 1, os usuários fazem menção a uma "lista de alvos" da Ucrânia e à ataques ucranianos a um escritor Russo. A Comunidade 3 têm usuários que criticam o uso de urânio empobrecido (fornecido à Ucrânia para uso militar pelos EUA) e alegam uso de simbologia nazista em tanques ucranianos. De maneira geral, esse *subreddit* se destaca por aparentar não ser pró-Ucrânia, diferentemente dos outros analisados neste trabalho.

Figura 46 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Leiden em r/EndlessWar.

(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

Figura 47 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/EndlessWar.



Fonte: Elaboração própria.

5.3.2.2 r/RussiaUkraineWar2022

A Tabela 18 mostra o grafo das estatísticas encontradas para as 10 maiores Comunidades detectadas pelo algoritmo de Leiden em "r/RussiaUkraineWar2022". O número de nodos decresce rapidamente das duas primeiras Comunidades para as outras. Todas as Comunidades a partir da Comunidade 4 tiveram um tamanho relativamente próximo, ainda que o número de arestas seja mais variável entre elas. O grau médio também não foi proporcional ao tamanho de cada Comunidade, mas o valor desse dado em todas as Comunidades desse *subreddit* foi relativamente elevado, comparando-o às estatísticas de outros *subreddits*. O diâmetro de todas as Comunidades foi extremamente parecido.

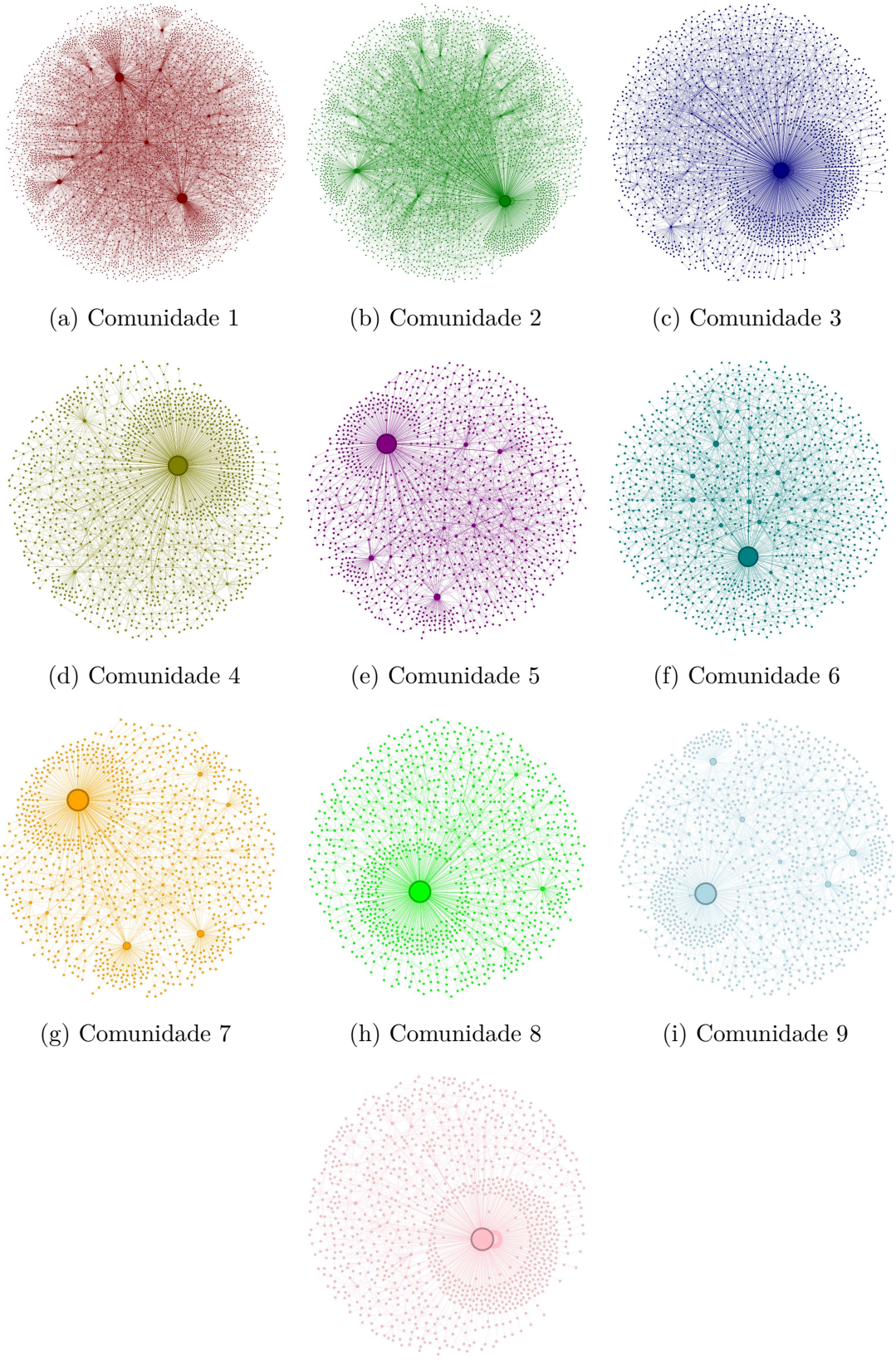
Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	3562	6887	3,86	515	10
2	2794	5028	3,59	928	9
3	1767	2553	2,89	1019	10
4	1205	1656	2,74	661	10
5	1140	1675	2,93	317	9
6	1050	1636	3,11	208	11
7	1034	1343	2,59	396	10
8	1007	1304	2,59	434	11
9	997	1425	2,85	287	11
10	997	1299	2,6	533	9
Todas	15553	42617	5,48	3209	Não calculado

Tabela 18 – Dados das Comunidades encontradas para r/RussiaUkraineWar2022.

Observando a Figura 48, que mostra os grafos gerados a partir de cada Comunidade, é perceptível que algumas Comunidades tiveram uma dispersão do grau dentre a extensão da rede, com mais *hubs* aparecendo em cada grafo. Isso também é perceptível comparando estas visualização com a Tabela 18, onde vemos um grau médio bem alto em Comunidades que possuem diversos *hubs*, como a 1 e a 2. A Comunidade 3, por sua vez, parece centrada em apenas um usuário, e como consequência teve um grau máximo muito alto mas um grau médio menor.

Analisando o grafo de todas as Comunidades unidas, apresentado na Figura 49, é notável um comportamento diferente dos grafos referentes aos *subreddits* anteriores. Dessa vez, as regiões correspondentes às Comunidades 1 e 2 ficaram bem destacadas em polos opostos do grafo, sinal de que estas duas Comunidades não possuem muita interação entre seus usuários. Uma possível conclusão disso é que essas duas Comunidades na verdade estão compostas por usuários que interagiram dentro do *subreddit* em eventos que aconteceram em momentos completamente diferentes do conflito Russo-Ucraniano. Assim, esses usuários podem não ter interações uns com os outros simplesmente por estarem ativos no fórum em períodos de tempo diferentes.

Figura 48 – Visualização dos grafos das Comunidades de r/RussiaUkraineWar2022.



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

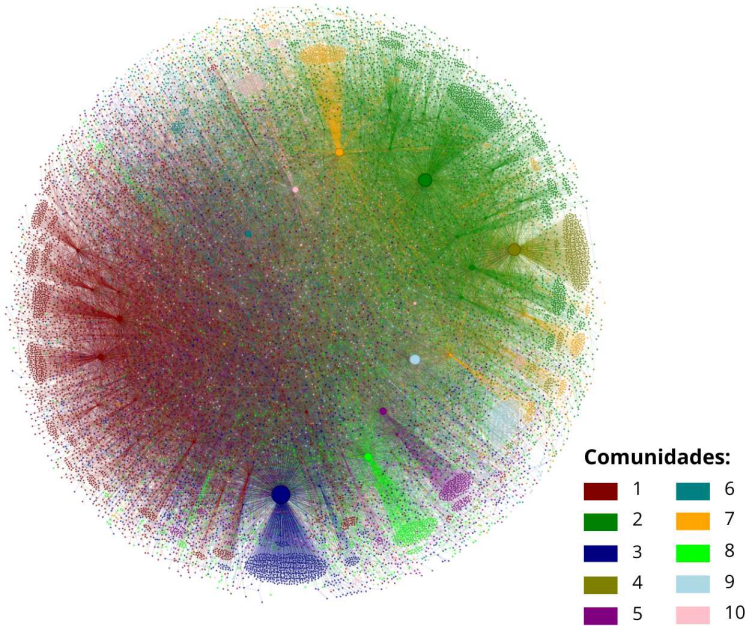
(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

(g) Comunidade 7 (h) Comunidade 8 (i) Comunidade 9

(j) Comunidade 10

Fonte: Elaboração própria.

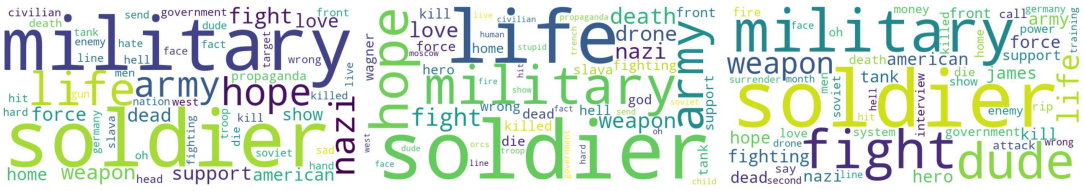
Figura 49 – Grafo da junção das Comunidades de r/RussiaUkraineWar2022, detectadas pelo algoritmo de Leiden.



Fonte: Elaboração própria.

A Figura 50 mostra as nuvens de palavras geradas a partir das três maiores Comunidades detectadas por Leiden em "r/RussiaUkraineWar2022". A Comunidade 2 em particular se destacou por ter palavras de conotação positiva em maior evidência. De forma geral, as três imagens foram bem parecidas.

Figura 50 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Leiden em r/RussiaUkraineWar2022.

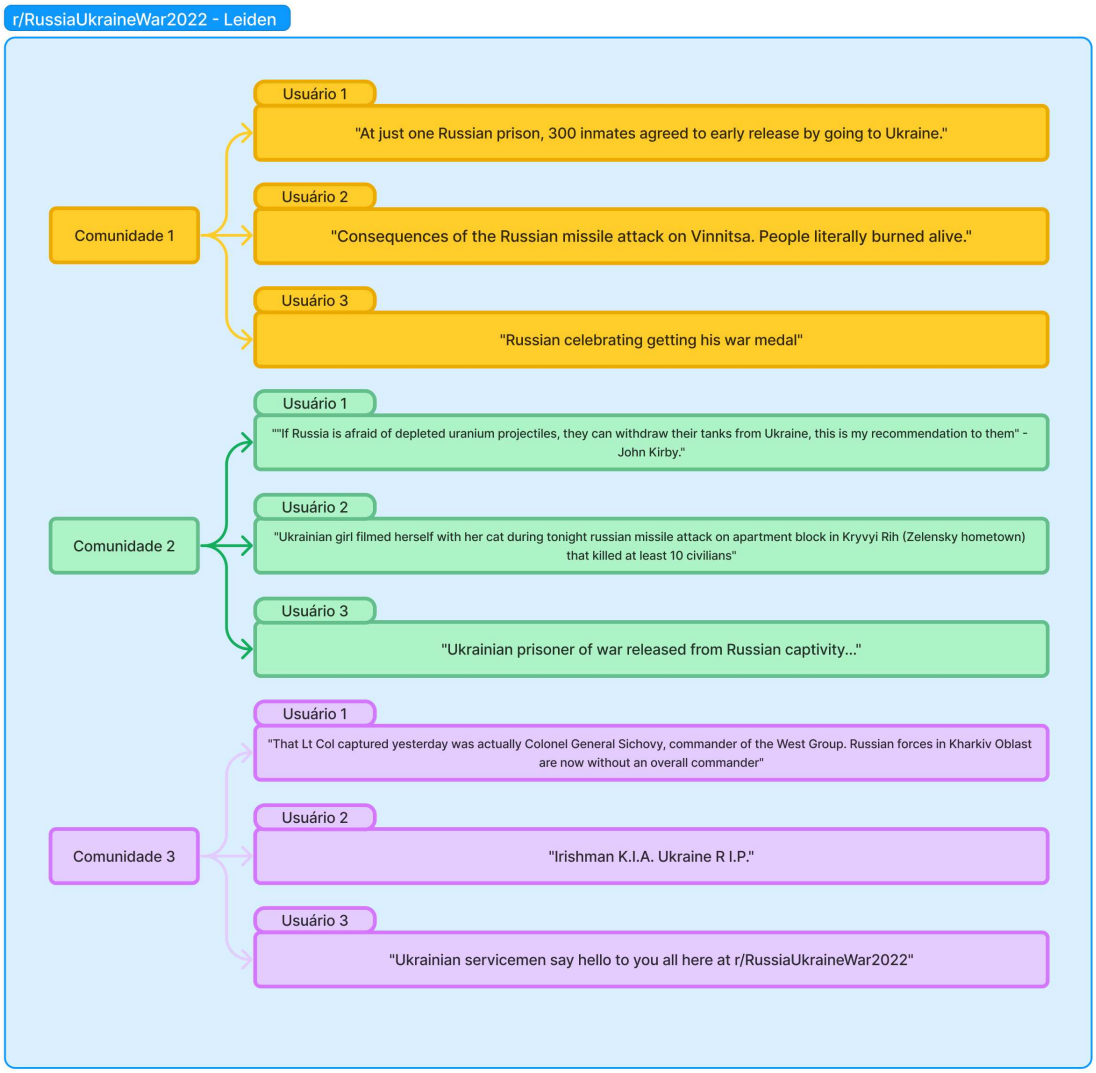


(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

A Figura 51 mostra as postagens ou comentários feitos pelos usuários de maior repercussão dentro de cada Comunidade. De forma geral, os comentários que foram destacadas parecem referenciar eventos muito prejudiciais ou muito positivos para o lado ucraniano da guerra. A Comunidade 1, por exemplo, contém uma postagem mostrando a destruição Russa na cidade de Vinnitsa, junto com outra notícia sobre prisioneiros ucranianos sendo libertados de uma prisão Russa. De maneira geral, esse *subreddit* parece apoiar fortemente a Ucrânia na guerra.

Figura 51 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/RussiaUkraineWar2022.



Fonte: Elaboração própria.

5.3.2.3 r/Ukraine

A Tabela 19 mostra as estatísticas das Comunidades encontradas pelo algoritmo de Leiden no *subreddit* "r/Ukraine". As 3 primeiras Comunidades obtiveram um grande número de nodos comparado a Comunidades de *subreddits* anteriores. O número de arestas em geral foi proporcional ao tamanho de cada Comunidade, com exceção das Comunidades 2 e 3. As 4 primeiras Comunidades também tiveram um grau médio bem elevado, um sinal de que as discussões que aconteceram nesse *subreddit* provavelmente possuíram uma quantidade acima da média de réplicas e tréplicas nos comentários, fazendo com que grande parte dos usuários fique com um grau maior de interações. A Comunidade 3, em particular, teve um valor de grau médio muito alto, combinado a um valor de grau máximo

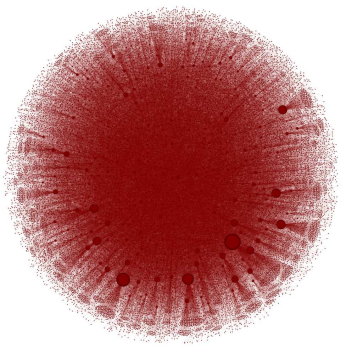
relativamente baixo comparado com as outras Comunidades. Isso implica que o grau médio não foi "puxado para cima" por alguns poucos usuários, mas que de maneira geral os usuários dessa Comunidade tiveram uma grande quantidade de interações.

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	18895	44449	4,7	837	11
2	13942	27213	3,9	1674	10
3	13217	36670	5,54	342	11
4	7796	14354	3,68	1365	11
5	7116	12437	3,49	494	13
6	5519	8211	2,97	347	12
7	3918	5008	2,25	718	14
8	3458	4421	2,55	336	14
9	1986	2276	2,29	698	14
10	1938	2272	2,34	358	14
Todas	77785	294389	7,56	3548	Não calculado

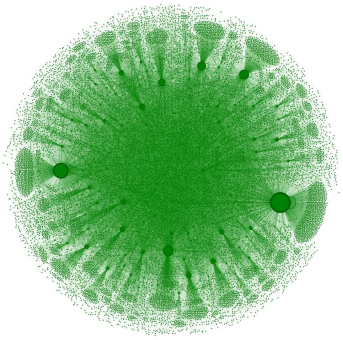
Tabela 19 – Dados das Comunidades encontradas para r/Ukraine.

A Figura 52 revela os grafos gerados a partir das Comunidades que foram detectadas pelo algoritmo de Leiden nesse *subreddit*. As 3 primeiras Comunidades ficaram visualmente bem densas, uma consequência da quantidade grande de número de nodos somado ao valor alto de grau médio, fazendo com que os nodos no geral tenham um tamanho maior no grafo. A Comunidade 3 em particular reflete isso, possuindo diversos *hubs* diferentes espalhados por toda a extensão da rede. A Comunidade 6 teve uma grande quantidade de *hubs*, mas o grau médio mais baixo revela que apesar desses usuários com grau mais elevado, também existe uma grande quantidade de usuários com grau mais reduzido. Uma possível interpretação desse comportamento é que essa Comunidade é constituída por usuários que receberam muitas respostas em suas postagens ou comentários, mas essas respostas não geraram mais repercussão.

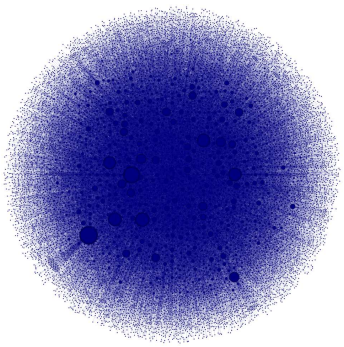
Figura 52 – Visualização dos grafos das Comunidades de r/Ukraine.



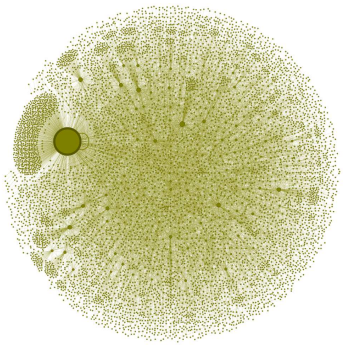
(a) Comunidade 1



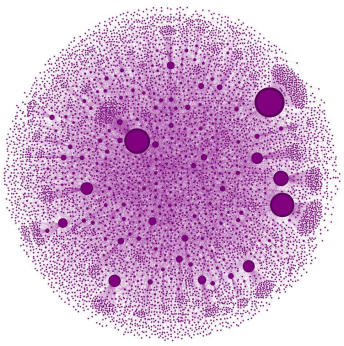
(b) Comunidade 2



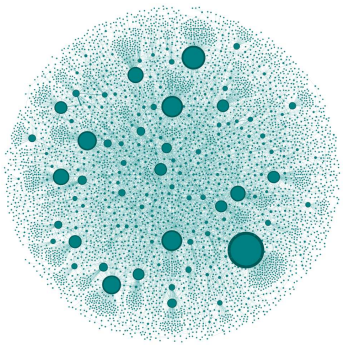
(c) Comunidade 3



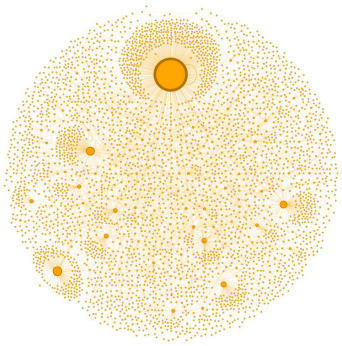
(d) Comunidade 4



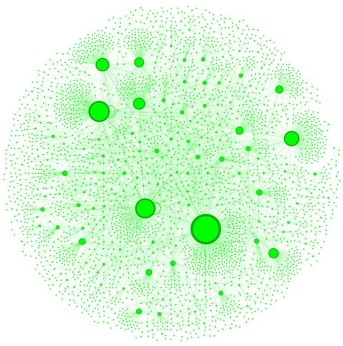
(e) Comunidade 5



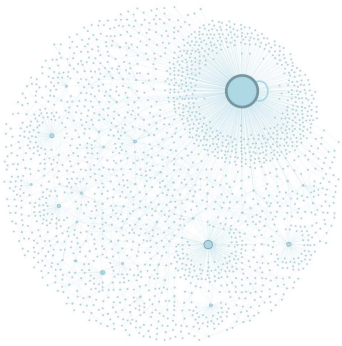
(f) Comunidade 6



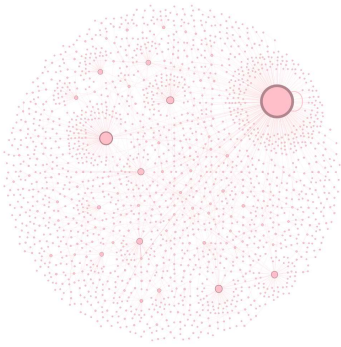
(g) Comunidade 7



(h) Comunidade 8



(i) Comunidade 9

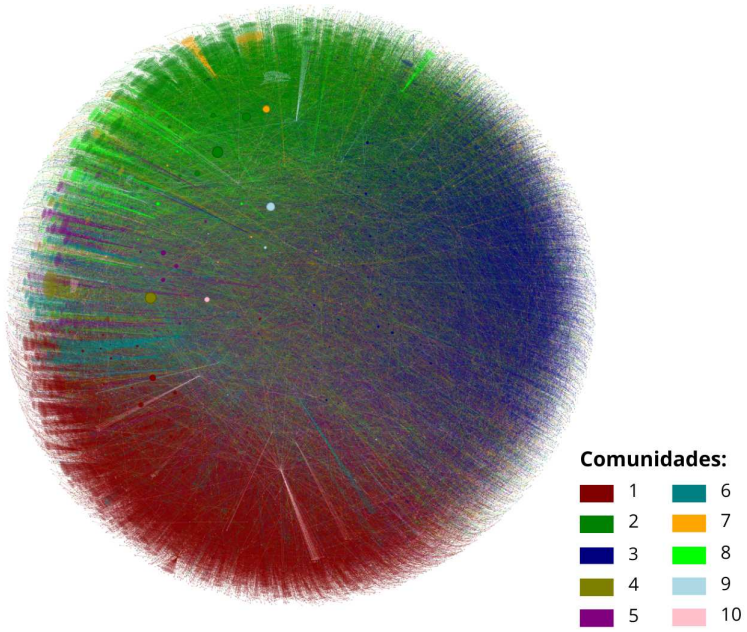


(j) Comunidade 10

Fonte: Elaboração própria.

O grafo da junção das Comunidades, disponível na Figura 53, revela que as 3 maiores Comunidades detectadas para esse *subreddit* dividiram o grafo em terços de tamanho muito similar, o que significa que não houve tanta interação entre os usuários presentes nelas. É possível que cada uma dessas Comunidades corresponda a momentos diferentes do conflito que tiveram grande impacto. Assim, cada Comunidade teve uma grande quantidade de usuários e interações, mas devido à diferença no intervalo de tempo dos eventos, os usuários não formaram interações com outros usuários de Comunidades diferentes.

Figura 53 – Grafo da junção das Comunidades de r/Ukraine, detectadas pelo algoritmo de Leiden.

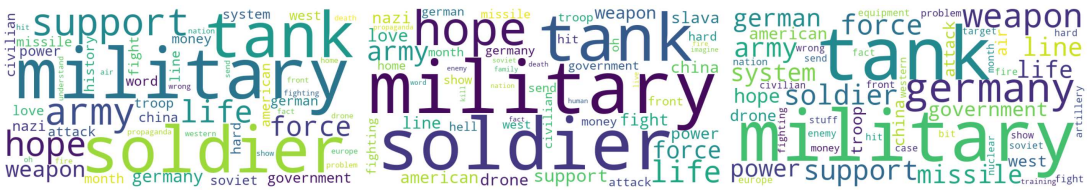


Fonte: Elaboração própria.

A Figura 54 mostra as nuvens de palavras geradas a partir das três maiores Comunidades. Novamente, as três Comunidades parecem ser bem similares em relação ao conteúdo sendo discutido, e a maior parte das palavras destacadas são de contexto militar.

A Figura 55 mostra as postagens ou comentários de maior repercussão dentro destas três Comunidades. Todos revelam um ponto de vista pró-Ucrânia, como esperado dentro deste *subreddit*. Parece existir uma tendência, observada em todos os *subreddits* apoiadores da Ucrânia, de que postagens ou comentários obtenham grande repercussão quando expõem eventos da guerra de uma perspectiva mais "pessoal". Isso é evidenciado na Comunidade 2, que apresenta menções sobre as atividades dos soldados ucranianos. Postagens que destacam grandes perdas da Rússia também recebem bastante atenção, como o comentário do usuário 3 na Comunidade 3, que documenta o número de baixas do exército russo até certa data.

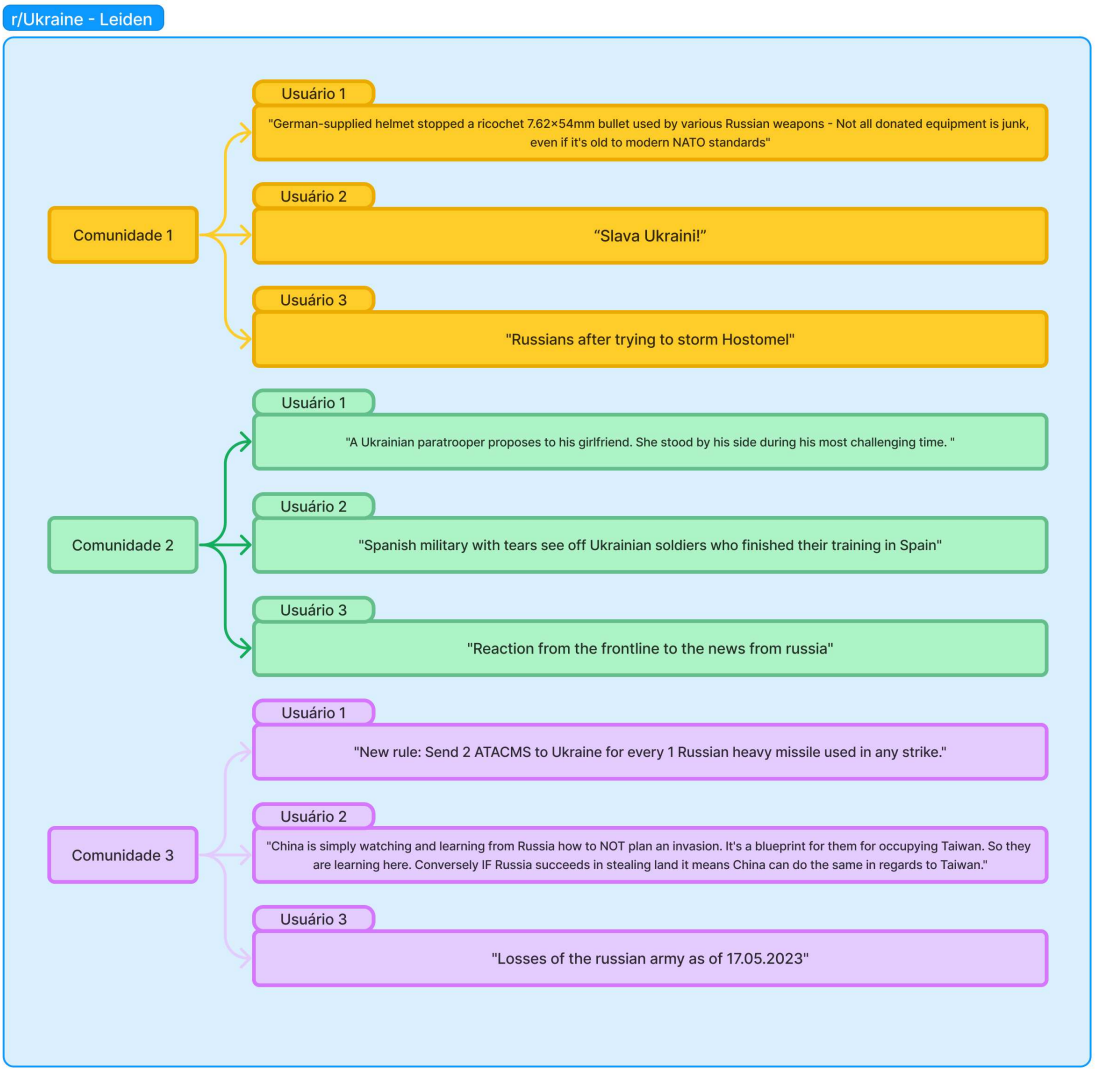
Figura 54 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Leiden em r/Ukraine.



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

Figura 55 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/Ukraine.



Fonte: Elaboração própria.

5.3.2.4 r/UkraineWarVideoReport

A Tabela 20 mostra as estatísticas das Comunidades detectadas pelo algoritmo de Leiden para o *subreddit* "r/UkraineWarVideoReport". Assim como o *subreddit* anterior, este também obteve valores bem altos de número de nodos, número de arestas e grau médio particularmente nas Comunidades 1, 2 e 3. As Comunidades 1, 4, 5 e 9 também tiveram um valor alto de grau máximo, indicando a presença de usuários que alcançaram muito destaque com suas postagens ou comentários. O diâmetro foi maior para Comunidades menores.

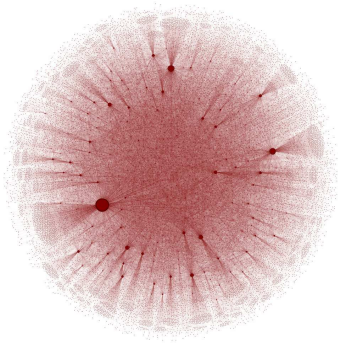
Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	15011	37738	5,02	2216	9
2	14129	28450	4	898	9
3	10085	20422	4,05	726	10
4	4847	7123	2,93	1970	11
5	3928	5516	2,8	1098	13
6	3790	5435	2,86	600	14
7	3784	5090	2,69	883	16
8	3615	4853	2,68	687	13
9	3460	4411	2,55	1018	14
10	3450	4825	2,79	698	13
Todas	66189	211348	6,38	5925	Não calculado

Tabela 20 – Dados das Comunidades encontradas para r/UkraineWarVideoReport.

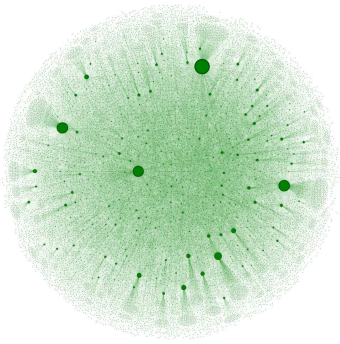
A Figura 56 revela os grafos que foram gerados a partir das Comunidades detectadas para este *subreddit*. As 3 primeiras Comunidades tiveram uma grande quantidade de *hubs* diferentes, o que é esperado dado seus valores elevados de grau médio. Por outro lado, as Comunidades 4 e 5 tiveram apenas um usuário de destaque, enquanto o resto do grafo possui nodos com grau mais baixo, o que explica os seus valores comparativamente menores de grau médio e ao mesmo tempo um alto grau máximo.

O grafo conjunto das Comunidades detectadas para esse *subreddit*, apresentado na Figura 57, mostra como as 4 maiores Comunidades dividiram a rede em 4 grandes regiões correspondentes. Pela posição dessas regiões, é possível inferir que a Comunidade 1 e 4 possuem certo grau de conectividade entre seus usuários, assim como as Comunidades 2 e 3. A posição mais central das Comunidades 4 e 5 também pode explicar os seus valores elevados de grau máximo, pois implica que os *hubs* dessas Comunidades estão conectados com nodos de diversas outras Comunidades.

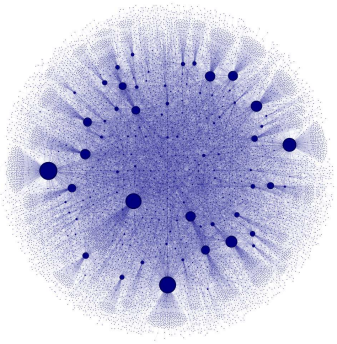
Figura 56 – Visualização dos grafos das Comunidades de r/UkraineWarVideoReport.



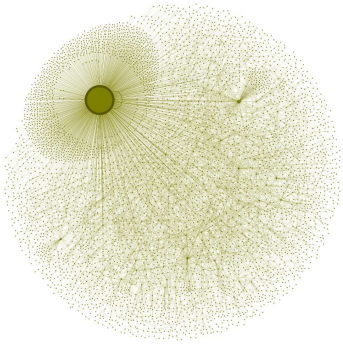
(a) Comunidade 1



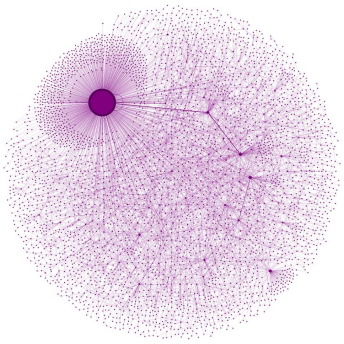
(b) Comunidade 2



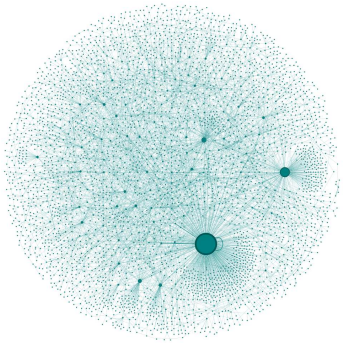
(c) Comunidade 3



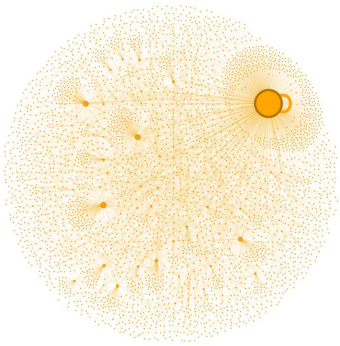
(d) Comunidade 4



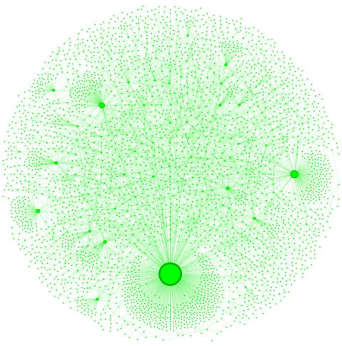
(e) Comunidade 5



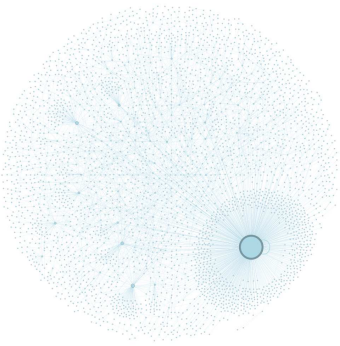
(f) Comunidade 6



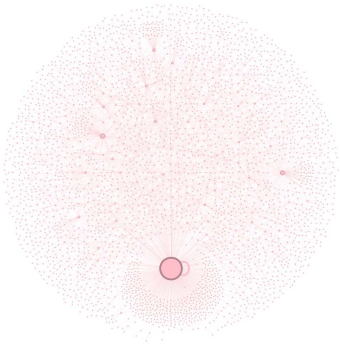
(g) Comunidade 7



(h) Comunidade 8



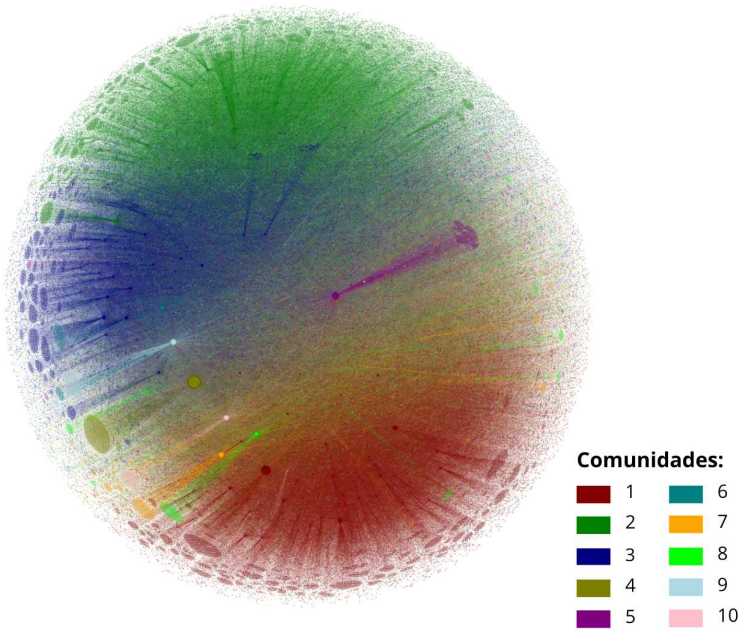
(i) Comunidade 9



(j) Comunidade 10

Fonte: Elaboração própria.

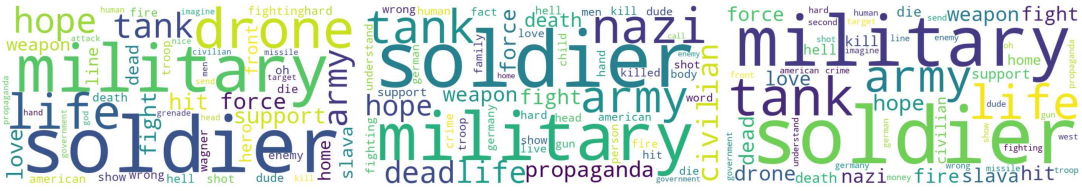
Figura 57 – Grafo da junção das Comunidades de r/UkraineWarVideoReport, detectadas pelo algoritmo de Leiden.



Fonte: Elaboração própria.

A Figura 58 mostra as nuvens de palavras geradas a partir das três maiores Comunidades detectadas por Leiden no *subreddit* "r/UkraineWarVideoReport". As três imagens destacam palavras relacionadas a conceitos militares, indicando que as três Comunidades possivelmente tiveram tópicos de discussão muito parecidos.

Figura 58 – Nuvens de palavras das três maiores Comunidades encontradas pelo algoritmo de Leiden em r/UkraineWarVideoReport.



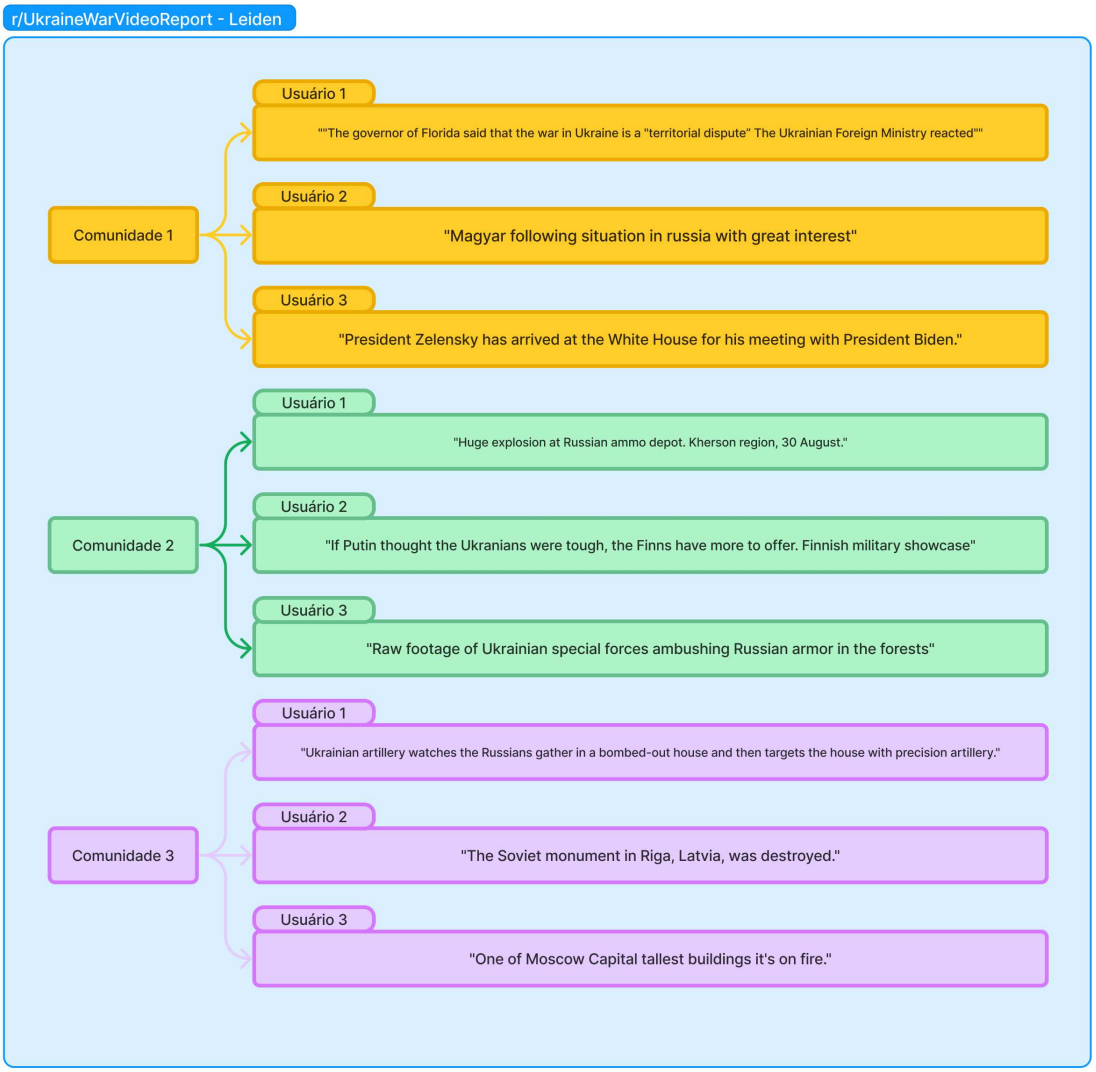
(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

Fonte: Elaboração própria.

A Figura 59 revela as postagens ou comentários de maior repercussão nessas três Comunidades. A Comunidade 3 em particular parece centrada em tópicos que destacam grandes perdas por parte da Rússia, mencionando a destruição de um monumento soviético e de um prédio da capital Moscou, e o bombardeamento realizado pela artilheria Ucraniana direcionado a russos. A Comunidade 2 evidência tópicos relacionados a questões militares, como operações do exército ucraniano e a perda de um depósito de munições russo. A Comunidade 1 parece ter foco político, fazendo menção ao encontro dos presidentes da

Ucrânia e dos EUA, e o interesse na guerra por parte do governador da Flórida na época e do político Húngaro Peter Magyar.

Figura 59 – Comentários de alta repercussão nas três maiores Comunidades detectadas em r/UkraineWarVideoReport.



Fonte: Elaboração própria.

5.4 COMPARAÇÃO ENTRE ALGORITMOS

Esta seção apresenta comparações diretas entre os resultados obtidos pelos algoritmos de Louvain e de Leiden, para facilitar a análise das diferenças entre os dois. A Tabela 21 compara os valores de modularidade que foram obtidos para cada um dos *subreddits*. Louvain obteve valores maiores em "r/EndlessWar" e "r/RussiaUkraineWar2022", enquanto Leiden obteve valores maiores nos outros dois *subreddits*. Entretanto, as diferenças foram bem pequenas entre os dois algoritmos, indicando que as Comunidades detectadas

por cada um deles são similares em relação ao quanto os nodos de cada Comunidade estão densamente conectados entre si.

Comunidade	Louvain	Leiden
r/EndlessWar	0,330	0,288
r/RussiaUkraineWar2022	0,346	0,335
r/Ukraine	0,345	0,362
r/UkraineWarVideoReport	0,393	0,401

Tabela 21 – Comparação de valores de modularidade obtidos pelos algoritmos de Louvain e Leiden

A Tabela 22 fornece a média de cada uma das estatísticas das 10 maiores Comunidades detectadas por cada algoritmo em "r/EndlessWar". O algoritmo de Louvain gerou Comunidades com uma média maior de número de nodos, mas com número de arestas similar e grau médio menor, mostrando que suas Comunidades tem, em média, uma densidade de interações menor. Curiosamente, o grau máximo obtido por Louvain ainda assim foi maior em média, possivelmente sinalizando que as interações dentro de suas Comunidades foram mais centradas em um usuário específico. O diâmetro obtido pelos dois algoritmos foi similar.

Algoritmo	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
Louvain	196,7	274,1	2,731	78,2	7,4
Leiden	174,4	275,4	3,069	43,6	7

Tabela 22 – Comparação das médias das estatísticas obtidas por cada algoritmo para as 10 maiores Comunidades detectadas em r/EndlessWar

A Tabela 23 fornece a comparação das médias para "r/RussiaUkraineWar2022". Novamente, o algoritmo de Louvain teve uma média maior de número de nodos, e superou Leiden em média de número de arestas. Entretanto, a média de grau de Leiden ainda foi maior, indicando que apesar de em geral suas Comunidades terem menos usuários, cada usuário tende a receber ou realizar mais interações.

Algoritmo	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
Louvain	2050,2	3108,6	2,896	567,6	10,6
Leiden	1555,3	2480,6	2,975	529,8	10

Tabela 23 – Comparação das médias das estatísticas obtidas por cada algoritmo para as 10 maiores Comunidades detectadas em r/RussiaUkraineWar2022

A Tabela 24 fornece as médias para "r/Ukraine". Dessa vez, o algoritmo de Leiden detectou Comunidades em média maiores em número de nodos e arestas. Ainda assim, a tendência de cada Comunidade ter uma média de grau maior que aquelas detectadas por Louvain continua, mostrando que mesmo com mais usuários em cada Comunidade, o algoritmo de Leiden ainda detecta Comunidades mais densas em relação a quantidade de

interações de cada usuário. O grau máximo médio de Louvain ainda foi maior, possivelmente indicando que em suas Comunidades existe uma presença maior de usuários com um grau excepcionalmente alto relativo aos outros membros da Comunidade.

Algoritmo	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
Louvain	7591,7	12601,2	3,154	750,7	13,2
Leiden	7778,5	15731,1	3,371	716,9	12,4

Tabela 24 – Comparação das médias das estatísticas obtidas por cada algoritmo para as 10 maiores Comunidades detectadas em r/Ukraine

A Tabela 25, por sua vez, fornece as médias para "r/UkraineWarVideoReport". Os dois algoritmos obtiveram uma média extremamente similar de número de arestas nas Comunidades detectadas, mas Leiden possui um número de nodos menor e grau médio maior em média, novamente indicando uma densidade de interações maior entre seus nodos. Contudo, nesse *subreddit* o grau máximo médio também foi maior para Leiden, o que implica que seu grau médio possivelmente indica a existência de usuários com uma quantidade de interações muito alta nas Comunidades que foram detectadas. Isso também pode ter provocado uma influência grande no grau médio, de forma que a maior parte das interações em cada Comunidade tenha sido de responsabilidade dos usuários *hubs*.

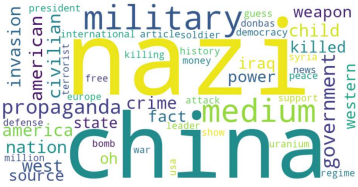
Algoritmo	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
Louvain	7019,9	12367,9	3,032	870	13,8
Leiden	6609,9	12386,3	3,237	1079,4	12,2

Tabela 25 – Comparação das médias das estatísticas obtidas por cada algoritmo para as 10 maiores Comunidades detectadas em r/UkraineWarVideoReport

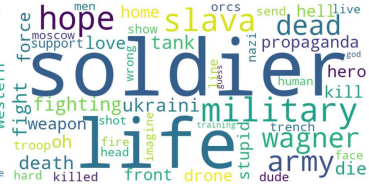
De maneira geral, o algoritmo de Louvain parece ter detectado Comunidades com a presença de *hubs* de grau muito elevado, o que explicaria a tendência de seu grau máximo ser em média maior que as Comunidades de Leiden. Contudo, o algoritmo de Leiden parece detectar Comunidades com grau médio constantemente maior, indicando que os usuários que populam cada Comunidade em geral interagem mais uns com os outros.

A seguir, nas figuras 60 e 61, é apresentada uma comparação entre as nuvens de palavras das maiores comunidades encontradas pelos dois algoritmos em cada *subreddit*, a fim de facilitar a comparação do conteúdo sendo discutido em cada fórum.

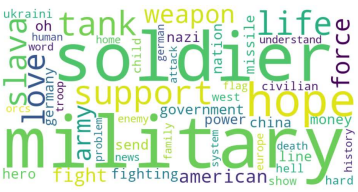
Figura 60 – Nuvens de palavras geradas a partir das maiores comunidades encontradas pelo algoritmo de Louvain em cada *subreddit*.



(a) r/EndlessWar



(b) r/RussiaUkraineWar2022



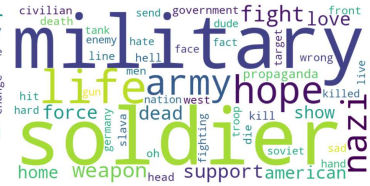
(c) r/Ukraine

(d) r/UkraineWarVideoReport

Fonte: Elaboração própria.

Figura 61 – Nuvens de palavras geradas a partir das maiores comunidades encontradas pelo algoritmo de Leiden em cada *subreddit*.

(a) r/EndlessWar



(b) r/RussiaUkraineWar2022

(c) r/Ukraine

(d) r/UkraineWarVideoReport

Fonte: Elaboração própria.

Ao comparar os dois algoritmos, é perceptivo que as comunidades encontradas pelos dois são bem similares em relação às palavras mais utilizadas, indicando que também pode existir grande semelhança entre os usuários participantes de cada comunidade.

Entretanto, comparando cada *subreddit* individualmente, é notável que a nuvem mais díspar é a de "r/EndlessWar" nos resultados de ambos os algoritmos. A fim de compreender melhor as diferenças entre o conteúdo de cada comunidade, a figura 62 mostra os comentários de maior repercussão encontrados nas comunidades detectadas pelos dois algoritmos em cada *subreddit*.

Figura 62 – Comparação entre comentários de maior repercussão nas maiores comunidades encontradas por cada algoritmo em cada *subreddit*

Fonte: Elaboração própria.

O *subreddit* "r/EndlessWar" novamente se destacou, pelo fato de ambos os comentários destacados possuírem uma visão mais neutra, ou até mesmo crítica para o lado ucraniano, criticando sua mídia e o uso de *fake news* durante a guerra. Todos os outros *subreddits* apresentaram comentários celebrando vitórias da Ucrânia ou derrotas da Rússia. Assim, é possível concluir que os *subreddits* analisados neste trabalho majoritariamente são pró-Ucrânia.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A etapa inicial deste trabalho consistiu na extração e posterior caracterização de dados, para auxiliar o objetivo futuro de aplicar algoritmos de detecção de comunidades. Para este fim, foram aplicadas algumas técnicas de análise, como a análise de sentimentos utilizando VADER e a análise psicolinguística utilizando LIWC, bem como a apresentação visual do conjunto de dados utilizando gráficos e nuvens de palavra.

As principais dificuldades nesta etapa do trabalho foram o estudo e aplicação das ferramentas de análise de sentimentos e psicolinguística, e também a elaboração de alguns gráficos apresentados, como as nuvens de palavras e a rede de co-ocorrência. Essas dificuldades surgiram, em parte, da necessidade de adaptar os dados extraídos do Reddit para que funcionassem com os algoritmos para análise e para gerar os gráficos.

Posteriormente, os dados coletados também foram utilizados para a geração de grafos, por meio do software Gephi. Os grafos gerados consistem de nodos representando os usuários de cada *subreddit*, e arestas representando interações entre esses usuários. A partir destes grafos foi possível extrair diversas estatísticas de cada *subreddit*, como o número de usuários participantes e quantas vezes cada usuários interagiu dentro da comunidade. Os resultados obtidos nessa etapa foram úteis para realizar análises dos comportamentos e fenômenos sociais ocorridos nestes fóruns de discussão.

Ainda com o auxílio do Gephi, e utilizando os algoritmos de detecção de comunidades de Louvain e de Leiden, foram gerados grafos das 10 maiores comunidades detectadas por cada algoritmo para os *subreddits* mais relevantes ao conflito entre a Rússia e a Ucrânia. Essas comunidades, por sua vez, também forneceram estatísticas importantes para compreender melhor as interações entre usuários e a interconectividade entre os participantes de cada comunidade. Além disso, foi possível comparar os resultados dos dois algoritmos executados e entender melhor suas semelhanças e diferenças.

O desafio principal desse ponto do trabalho foi compreender as implicações de cada estatística obtida a partir dos grafos gerados e da estrutura de cada grafo. Fatores como a quantidade de interações média dos usuários dentro de uma comunidade ou a posição de um nodo em um grafo possuem significados diferentes dependendo do contexto do *subreddit* de origem. Assim, além do estudo da teoria de grafos e do funcionamento dos algoritmos de detecção, também foi necessária uma compreensão do Reddit como ferramenta social, com uma cultura própria que influenciou todos os resultados obtidos.

A partir da comparação dos resultados ao final do desenvolvimento, ficou evidente que, de forma geral, as comunidades detectadas por cada algoritmo foram semelhantes em relação aos temas que mais foram discutidos e mais geraram repercussão, ainda que os usuários de maior grau em cada comunidade diferiram por algoritmo utilizado. Os *subreddits* que foram analisados neste trabalho se mostraram, em grande parte, apoiadores do lado ucraniano do conflito, com uma notável exceção sendo o *subreddit* “r/EndlessWar”,

que apresentou um ponto de vista mais neutro, criticando ambos os países em guerra.

Em relação a trabalhos futuros, cabem as seguintes propostas:

- Executar outros algoritmos de detecção de comunidades no conjunto de dados coletados;
- Identificar outros *subreddits* relevantes ao tópico deste trabalho que possam aumentar a diversidade do material analisado;
- Delimitar intervalos de tempo menores nos dados coletados, para realizar análises comparativas de períodos diferentes;
- Realizar análise de sentimentos e psico-linguística para as comunidades detectadas por cada algoritmo.

REFERÊNCIAS

ANDERSON, Katie. Ask me anything: what is Reddit? **Library Hi Tech News**, v. 32, p. 8–11, jul. 2015. DOI: 10.1108/LHTN-03-2015-0018. Disponível em: https://www.researchgate.net/publication/281391439_Ask_me_anything_what_is_Reddit.

BEDI, Punam; SHARMA, Chhavi. Community detection in social networks. **WIREs Data Mining and Knowledge Discovery**, v. 6, n. 3, p. 115–135, 2016. DOI: https://doi.org/10.1002/widm.1178. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1178. Disponível em: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1178.

BHUTKAR, Ganesh. Users on Social Networking Sites. **Journal of HCI Vistas**, v. V, fev. 2009. Disponível em: https://www.researchgate.net/publication/283664477_Users_on_Social_Networking_Sites.

BLONDEL, Vincent D *et al.* Fast unfolding of communities in large networks. **Journal of Statistical Mechanics: Theory and Experiment**, IOP Publishing, v. 2008, n. 10, p10008, out. 2008. ISSN 1742-5468. DOI: 10.1088/1742-5468/2008/10/p10008. Disponível em: http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

CARVALHO, Flavio *et al.* Evaluating the Brazilian Portuguese version of the 2015 LIWC Lexicon with sentiment analysis in social networks. *In:* p. 24–34. DOI: 10.5753/brasnam.2019.6545. Disponível em: https://sol.sbc.org.br/index.php/brasnam/article/view/6545.

CATALANO, Michael; LEISE, Tanya; PFAFF, Thomas. Measuring Resource Inequality: The Gini Coefficient. **Numeracy**, v. 2, jul. 2009. DOI: 10.5038/1936-4660.2.2.4.

CHOUDHURY, M.D.; DE, Sushovan. Mental Health Discourse on reddit: Self-Disclosure, Social Support, and Anonymity. **Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014**, v. 8, p. 71–80, mai. 2014. DOI: 10.1609/icwsm.v8i1.14526. Disponível em: https://ojs.aaai.org/index.php/ICWSM/article/view/14526.

CHUNAEV, Petr. Community detection in node-attributed social networks: A survey. **Computer Science Review**, v. 37, p. 100286, 2020. ISSN 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2020.100286. Disponível em: https://www.sciencedirect.com/science/article/pii/S1574013720303865.

CIURIAK, Dan. The Role of Social Media in Russia's War on Ukraine, abr. 2022. DOI: 10.2139/ssrn.4078863. Disponível em: https://ssrn.com/abstract=4078863.

ELENA-IULIA, Varga. The Importance Of Social Media. **Annals - Economy Series**, v. 6, p. 80–91, dez. 2018. Disponível em: https://ideas.repec.org/a/cbu/jrnlec/y2018v6p80-91.html.

FORTUNATO, Santo. Community detection in graphs. **Physics Reports**, v. 486, n. 3, p. 75–174, 2010. ISSN 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2009.11.002. Disponível em: https://www.sciencedirect.com/science/article/pii/S0370157309002841.

GLINIECKA, Martyna. The Ethics of Publicly Available Data Research: A Situated Ethics Framework for Reddit. **Social Media + Society**, v. 9, n. 3, p. 20563051231192021, 2023. DOI: 10.1177/20563051231192021. eprint: https://doi.org/10.1177/20563051231192021. Disponível em: https://doi.org/10.1177/20563051231192021.

GUERRA, A; KARAKUŞ, O. Sentiment analysis for measuring hope and fear from Reddit posts during the 2022 Russo-Ukrainian conflict. **Frontiers in Artificial Intelligence**, v. 6, p. 1163577, 2023. DOI: 10.3389/frai.2023.1163577.

HE, Jialin; CHEN, Duanbing. A fast algorithm for community detection in temporal network. **Physica A: Statistical Mechanics and its Applications**, v. 429, p. 87–94, 2015. ISSN 0378-4371. DOI: https://doi.org/10.1016/j.physa.2015.02.069. Disponível em: https://www.sciencedirect.com/science/article/pii/S0378437115001922.

HINTZ, Elizabeth A.; BETTS, Timothy. Reddit in communication research: current status, future directions and best practices. **Annals of the International Communication Association**, Routledge, v. 46, n. 2, p. 116–133, 2022. DOI: 10.1080/23808985.2022.2064325. eprint: https://doi.org/10.1080/23808985.2022.2064325. Disponível em: https://doi.org/10.1080/23808985.2022.2064325.

HORNE, Benjamin D.; ADALI, Sibel; SIKDAR, Sujoy. **Identifying the social signals that drive online discussions: A case study of Reddit communities**. [*S.l.*: *s.n.*], 2017. arXiv: 1705.02673 [cs.SI]. Disponível em: https://arxiv.org/abs/1705.02673.

HUTTO, C.; GILBERT, Eric. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. **Proceedings of the International AAAI Conference on Web and Social Media**, v. 8, n. 1, p. 216–225, mai. 2014. DOI: 10.1609/icwsm.v8i1.14550. Disponível em: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

JACOMY, Mathieu *et al.* ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. **PLOS ONE**, Public Library of Science, v. 9, n. 6, p. 1–12, jun. 2014. DOI: 10.1371/journal.pone.0098679. Disponível em: https://doi.org/10.1371/journal.pone.0098679.

KRUSKAL, William H.; WALLIS, W. Allen. Use of Ranks in One-Criterion Variance Analysis. **Journal of the American Statistical Association**, [American Statistical Association, Taylor Francis, Ltd.], v. 47, n. 260, p. 583–621, 1952. ISSN 01621459. Disponível em: http://www.jstor.org/stable/2280779. Acesso em: 6 nov. 2023.

NEWMAN, M. E. J.; GIRVAN, M. Finding and evaluating community structure in networks. **Phys. Rev. E**, American Physical Society, v. 69, p. 026113, 2 fev. 2004. DOI: 10.1103/PhysRevE.69.026113. Disponível em: https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

PROFERES, Nicholas *et al.* Studying Reddit: A Systematic Overview of Disciplines, Approaches, Methods, and Ethics. **Social Media + Society**, v. 7, n. 2, p. 20563051211019004, 2021. DOI: 10.1177/20563051211019004. eprint: https://doi.org/10.1177/20563051211019004. Disponível em: https://doi.org/10.1177/20563051211019004.

TAUSCZIK, Yla R.; PENNEBAKER, James W. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. **Journal of Language and Social Psychology**, v. 29, n. 1, p. 24–54, 2010. DOI: 10.1177/0261927X09351676. Disponível em: https://doi.org/10.1177/0261927X09351676.

TRAAG, Vincent A; WALTMAN, Ludo; ECK, Nees Jan van. From Louvain to Leiden: guaranteeing well-connected communities. **Scientific reports**, v. 9, n. 1, p. 5233, 2019. DOI: 10.1038/s41598-019-41695-z. Disponível em: https://doi.org/10.1038/s41598-019-41695-z.

WELLMAN, Barry. An Electronic Group Is Virtually a Social Network. **S. Kiesler (org.), Culture of Internet (pp. 179-205). Hilsdale, NJ: Lawrence Erlbaum**, 1997. Disponível em: https://www.researchgate.net/publication/2359189_An_Electronic_Group_is_Virtually_a_Social_Network.

ZENHA, Luciana. Redes sociais online: o que são as redes sociais e como se organizam? **n. 49: Caderno de Educação**, 2018. Disponível em: https://revista.uemg.br/index.php/cadernodeeducacao/article/view/2809.

ZHU, Yiming *et al.* **A Reddit Dataset for the Russo-Ukrainian Conflict in 2022**. [*S.l.*: *s.n.*], 2022. arXiv: 2206.05107 [cs.SI].

APÊNDICE A – CÓDIGO

```
from matplotlib import pyplot as plt
import matplotlib.dates as mdates
import pandas as pd
import json
import pytz
import datetime
import os
import sys
import re
import nltk
from nltk.corpus import stopwords
from nltk.util import bigrams
from nltk.stem import WordNetLemmatizer
from unwanted_words import unwanted_words
from unwanted_words import selected_words_to_remove
from global_vars import *
from collections import Counter
from wordcloud import WordCloud
from liwc import LIWC, attribute_translation
from scipy.stats import kruskal
import numpy as np
import itertools
import networkx as nx
from matplotlib.colors import LogNorm
import utils
import logging
import time
from cdlib import algorithms as cd_lib_algos
from communities import algorithms as communities_algos

stop_words = stopwords.words('english')
stop_words.extend(unwanted_words)
stop_words.extend(selected_words_to_remove)
number_list = list(range(1, 101))  # Generates numbers from 1 to 100
string_list = [str(num) for num in number_list]  # Converts numbers to
    strings
stop_words.extend(string_list)
liwc = LIWC('LIWC2007_English100131.dic')
lemmatizer = WordNetLemmatizer()
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(
    levelname)s - %(message)s')

def findDateInterval(data, isReply):
    if isReply:
        keyToIterate = 'Comment Date'
```

```
    else:
        keyToIterate = 'Time'

    if data[keyToIterate] < 1645704000:
        return 1643037767
    elif data[keyToIterate] >= 1645704000 and data[keyToIterate] <
       1648123200:
        return 1645704000
    elif data[keyToIterate] >= 1648123200 and data[keyToIterate] <
       1650801600:
        return 1648123200
    elif data[keyToIterate] >= 1650801600 and data[keyToIterate] <
       1653393600:
        return 1650801600
    elif data[keyToIterate] >= 1653393600 and data[keyToIterate] <
       1656072000:
        return 1653393600
    elif data[keyToIterate] >= 1656072000 and data[keyToIterate] <
       1658664000:
        return 1656072000
    elif data[keyToIterate] >= 1658664000 and data[keyToIterate] <
       1661342400:
        return 1658664000
    elif data[keyToIterate] >= 1661342400 and data[keyToIterate] <
       1664020800:
        return 1661342400
    elif data[keyToIterate] >= 1664020800 and data[keyToIterate] <
       1666612800:
        return 1664020800
    elif data[keyToIterate] >= 1666612800 and data[keyToIterate] <
       1669291200:
        return 1666612800
    elif data[keyToIterate] >= 1669291200 and data[keyToIterate] <
       1671883200:
        return 1669291200
    elif data[keyToIterate] >= 1671883200 and data[keyToIterate] <
       1674561600:
        return 1671883200
    elif data[keyToIterate] >= 1674561600 and data[keyToIterate] <
       1677240000:
        return 1674561600
    elif data[keyToIterate] >= 1677240000 and data[keyToIterate] <
       1679670000:
        return 1677240000
    elif data[keyToIterate] >= 1679670000 and data[keyToIterate] <
       1682348400:
        return 1679670000
```

```
    elif data[keyToIterate] >= 1682348400 and data[keyToIterate] <
       1684940400:
        return 1682348400
    elif data[keyToIterate] >= 1684940400 and data[keyToIterate] <
       1687618800:
        return 1684940400
    else:
        return 1687618800

def iterateThroughPosts(dic, targetDict, targetDictReplies):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            targetDict[dateInterval] += 1
            iterateThroughPosts(data['Comment List'], targetDict,
               targetDictReplies)
        except Exception:
            dateInterval= findDateInterval(data, True)
            targetDictReplies[dateInterval] += 1
            iterateThroughPosts(data['Comment Replies'], targetDict,
               targetDictReplies)

def iterateToCountUsers(dic, targetDict, targetDictReplies,
   checkedUserDict):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            if data['Author'] not in checkedUserDict[dateInterval]:
                checkedUserDict[dateInterval].append(data['Author'])
                targetDict[dateInterval] += 1
            iterateToCountUsers(data['Comment List'], targetDict,
               targetDictReplies, checkedUserDict)
        except Exception:
            dateInterval= findDateInterval(data, True)
            if data['Author'] not in checkedUserDict[dateInterval]:
                checkedUserDict[dateInterval].append(data['Author'])
                targetDictReplies[dateInterval] += 1
            iterateToCountUsers(data['Comment Replies'], targetDict,
               targetDictReplies, checkedUserDict)

def iterateToCountPositiveOrNegative(dic, positiveWordCountDict,
   negativeWordCountDict):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            if data['Sentiment Label'] == 'Positive':
                positiveWordCountDict[dateInterval] += 1
```

```
            elif data['Sentiment Label'] == 'Negative':
                negativeWordCountDict[dateInterval] += 1
            iterateToCountPositiveOrNegative(data['Comment List'],
               positiveWordCountDict, negativeWordCountDict)
        except Exception:
            dateInterval= findDateInterval(data, True)
            if data['Sentiment Label'] == 'Positive':
                positiveWordCountDict[dateInterval] += 1
            elif data['Sentiment Label'] == 'Negative':
                negativeWordCountDict[dateInterval] += 1
            iterateToCountPositiveOrNegative(data['Comment Replies'],
               positiveWordCountDict, negativeWordCountDict)

def iterateToCountPositiveWords(dic, positiveCountDict,
   positiveCountDictReplies):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            if data['Sentiment Label'] == 'Positive':
                positiveCountDict[dateInterval] += 1
            iterateToCountPositiveWords(data['Comment List'])
        except Exception:
            dateInterval= findDateInterval(data, True)
            if data['Sentiment Label'] == 'Positive':
                positiveCountDictReplies[dateInterval] += 1
            iterateToCountPositiveWords(data['Comment Replies'])

def iterateToCountNegativeWords(dic, negativeCountDict,
   negativeCountDictReplies):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            if data['Sentiment Label'] == 'Negative':
                negativeCountDict[dateInterval] += 1
            iterateToCountNegativeWords(data['Comment List'])
        except Exception:
            dateInterval= findDateInterval(data, True)
            if data['Sentiment Label'] == 'Negative':
                negativeCountDictReplies[dateInterval] += 1
            iterateToCountNegativeWords(data['Comment Replies'])

def iterateThroughPostsKeywords(dic, keyword, targetDict):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            if keyword not in targetDict[dateInterval]:
                targetDict[dateInterval][keyword] = 1
```

```
            if keyword in data['Title'].lower():
                if keyword in targetDict[dateInterval]:
                    targetDict[dateInterval][keyword] += 1
                iterateThroughPostsKeywords(data['Comment List'],
                    keyword, targetDict)
        except Exception:
            dateInterval= findDateInterval(data, True)
            if keyword not in targetDict[dateInterval]:
                targetDict[dateInterval][keyword] = 1
            if keyword in data['Comment Content'].lower():
                if keyword in targetDict[dateInterval]:
                    targetDict[dateInterval][keyword] += 1
                iterateThroughPostsKeywords(data['Comment Replies'],
                    keyword, targetDict)

def iterateThroughPostsSentimentPolarity(dic, globalDic,
    globalDicReplies):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            globalDic[dateInterval] += data['Polarity Score']
            iterateThroughPostsSentimentPolarity(data['Comment List'],
                globalDic, globalDicReplies)
        except Exception:
            dateInterval= findDateInterval(data, True)
            if 'Comment Count' in list(globalDicReplies[dateInterval].
                keys()):
                globalDicReplies[dateInterval]['Comment Count'] += 1
                globalDicReplies[dateInterval]['Total Polarity'] += data
                    ['Polarity Score']
                iterateThroughPostsSentimentPolarity(data['Comment
                    Replies'], globalDic, globalDicReplies)
            else:
                globalDicReplies[dateInterval]['Comment Count'] = 0
                globalDicReplies[dateInterval]['Total Polarity'] = 0
                iterateThroughPostsSentimentPolarity(data['Comment
                    Replies'], globalDic, globalDicReplies)


def iterateThroughPostsSentimentPolarityKeywords(dic, keyword,
    targetDict):
    for data in dic:
        try:
            dateInterval = findDateInterval(data, False)
            # if keyword not in important_dates_keywords_sp[dateInterval
                ]:
```

```
            #     important_dates_keywords_sp[dateInterval][keyword] =
                {'keywordPolarityTotal': 0, 'keywordCount': 1}
            if keyword not in targetDict[dateInterval]:
                targetDict[dateInterval][keyword] = np.nan
            if keyword in data['Title'].lower():
                if targetDict[dateInterval][keyword] is np.nan:
                    targetDict[dateInterval][keyword] = {'
                        keywordPolarityTotal': 0, 'keywordCount': 1}
                #if keyword in targetDict[dateInterval]:
                targetDict[dateInterval][keyword]['keywordPolarityTotal'
                    ] += data['Polarity Score']
                targetDict[dateInterval][keyword]['keywordCount'] += 1
                iterateThroughPostsSentimentPolarityKeywords(data['
                    Comment List'], keyword, targetDict)
        except Exception:
            dateInterval= findDateInterval(data, True)
            if keyword not in targetDict[dateInterval]:
                targetDict[dateInterval][keyword] = np.nan
            if keyword in data['Comment Content'].lower():
                if targetDict[dateInterval][keyword] is np.nan:
                    targetDict[dateInterval][keyword] = {'
                        keywordPolarityTotal': 0, 'keywordCount': 1}
                #if keyword in targetDict[dateInterval]:
                targetDict[dateInterval][keyword]['keywordPolarityTotal'
                    ] += data['Polarity Score']
                targetDict[dateInterval][keyword]['keywordCount'] += 1
                iterateThroughPostsSentimentPolarityKeywords(data['
                    Comment Replies'], keyword, targetDict)

def iterateToGenerateThreadGraph(data, graph):
    for userInteraction in data:
        try:
            for userInteractionChild in userInteraction['Comment List']:
                graph.add_edge(userInteraction['ID'],
                    userInteractionChild['ID'])
                iterateToGenerateThreadGraph(userInteraction['Comment
                    List'], graph)
        except Exception:
            for userInteractionChild in userInteraction['Comment Replies
                ']:
                graph.add_edge(userInteraction['ID'],
                    userInteractionChild['ID'])
                iterateToGenerateThreadGraph(userInteraction['Comment
                    Replies'], graph)

id_list = []
```

```
def iterateToGenerateUserConnectionGraph(data, graph):
    for userInteraction in data:
        if userInteraction['ID'] not in id_list:
            id_list.append(userInteraction['ID'])
            try:
                for userInteractionChild in userInteraction['Comment
                    List']:
                    if (graph.has_edge(userInteraction['Author'],
                        userInteractionChild['Author'])):
                        graph[userInteraction['Author']][
                            userInteractionChild['Author']]['weight'] +=
                            1
                    else:
                        graph.add_edge(userInteraction['Author'],
                            userInteractionChild['Author'], weight=1)
                    iterateToGenerateUserConnectionGraph(userInteraction
                        ['Comment List'], graph)
            except Exception:
                for userInteractionChild in userInteraction['Comment
                    Replies']:
                    if (graph.has_edge(userInteraction['Author'],
                        userInteractionChild['Author'])):
                        graph[userInteraction['Author']][
                            userInteractionChild['Author']]['weight'] +=
                            1
                    else:
                        graph.add_edge(userInteraction['Author'],
                            userInteractionChild['Author'], weight=1)
                    iterateToGenerateUserConnectionGraph(userInteraction
                        ['Comment Replies'], graph)



def countCommentsByDate(file):
    dateDict = {
        1643037767: 0,
        1645704000: 0, # Data do in cio da guerra.
        1648123200: 0, # Mar o - Russia toma Kherson e Mariupol
        1650801600: 0, # Abril - Russia recua de Kyev e foca no leste.
            Alguns ataques brutais da Russia.
        1653393600: 0, # Maio - Azovstal - metal rgica s mbolo de
            resistencia    tomada.
        1656072000: 0, # Junho - For as Russas saem de Snake Island.
            V rios ataques de m sseis russos.
        1658664000: 0, # Julho - Acordo de gr os do Mar Negro,
            prisioneiros ucranianos assasinados.
```

```
        1661342400: 0, # Agosto - Contra ataque ucraninao em base a rea
            Russa.
        1664020800: 0, # Setembro - Russos fogem da regi o de Kharkiv,
           Ucr nia retoma uma grande  rea .
        1666612800: 0, # Outubro - Russia come a a alvejar a energia
           el trica da Ucr ncia com ataques a reos. M ssies em Kyev.
            Explos o na ponte Crimea.
        1669291200: 0, # Novembro - Kherson    liberada depois de oito
           meses. Blackout massivo.
        1671883200: 0, # Dezembro - Russia se torna o pa s mais
           sancionado. 100000 soldados russos perdidos. Zelensky visita
           USA.
        1674561600: 0, # Janeiro - Alemanha concorda em enviar tanques.
        1677240000: 0, # Fevereiro - Biden visita Kyev.
        1679670000: 0, # Mar o
        1682348400: 0, # Abril
        1684940400: 0, # Maio
        1687618800: 0 # Junho
    }

    dateDictCopy = dateDict.copy()


    with open(f'./data/jsonSubs/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateThroughPosts(data, dateDict, dateDictCopy)

    return dateDict, dateDictCopy

def countKeywordAppearancesByDate(file, keyword, targetDict):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateThroughPostsKeywords(data, keyword, targetDict)

def countSentimentPolarityByKeywordDate(file, keyword, targetDict):
    with open(f'./data/sentimentpolarity/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateThroughPostsSentimentPolarityKeywords(data, keyword,
           targetDict)

def countSentimentPolarityByDate(file, globalDic, globalDicReplies):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateThroughPostsSentimentPolarity(data, globalDic,
           globalDicReplies)
```

```
def countPositiveWords(file, positiveCountDict, positiveCountDictReplies
    ):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToCountPositiveWords(data, positiveCountDict,
            positiveCountDictReplies)

def countNegativeWords(file, negativeCountDict, negativeCountDictReplies
    ):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToCountNegativeWords(data, negativeCountDict,
            negativeCountDictReplies)

def countNumberOfUsersPerMonth(file):
    dateDict = {
        1643037767: 0,
        1645704000: 0, # Data do in cio da guerra.
        1648123200: 0, # Mar o - Russia toma Kherson e Mariupol
        1650801600: 0, # Abril - Russia recua de Kyev e foca no leste.
            Alguns ataques brutais da Russia.
        1653393600: 0, # Maio - Azovstal - metal rgica s mbolo de
            resistencia    tomada.
        1656072000: 0, # Junho - For as Russas saem de Snake Island.
            V rios ataques de m sseis russos.
        1658664000: 0, # Julho - Acordo de gr os do Mar Negro,
            prisioneiros ucranianos assasinados.
        1661342400: 0, # Agosto - Contra ataque ucraninao em base a rea
             Russa.
        1664020800: 0, # Setembro - Russos fogem da regi o de Kharkiv,
            Ucr nia retoma uma grande  rea .
        1666612800: 0, # Outubro - Russia come a a alvejar a energia
            el trica da Ucr ncia com ataques a reos. M ssies em Kyev.
             Explos o na ponte Crimea.
        1669291200: 0, # Novembro - Kherson    liberada depois de oito
            meses. Blackout massivo.
        1671883200: 0, # Dezembro - Russia se torna o pa s mais
            sancionado. 100000 soldados russos perdidos. Zelensky visita
            USA.
        1674561600: 0, # Janeiro - Alemanha concorda em enviar tanques.
        1677240000: 0, # Fevereiro - Biden visita Kyev.
        1679670000: 0, # Mar o
        1682348400: 0, # Abril
        1684940400: 0, # Maio
        1687618800: 0 # Junho
    }
```

```
    dateDictCopy = dateDict.copy()

    checkedUsersDict = {
        1643037767: [],
        1645704000: [], # Data do in cio da guerra.
        1648123200: [], # Mar o - Russia toma Kherson e Mariupol
        1650801600: [], # Abril - Russia recua de Kyev e foca no leste.
           Alguns ataques brutais da Russia.
        1653393600: [], # Maio - Azovstal - metal rgica s mbolo de
           resistencia    tomada.
        1656072000: [], # Junho - For as Russas saem de Snake Island.
           V rios ataques de m sseis russos.
        1658664000: [], # Julho - Acordo de gr os do Mar Negro,
           prisioneiros ucranianos assasinados.
        1661342400: [], # Agosto - Contra ataque ucraninao em base
           a rea Russa.
        1664020800: [], # Setembro - Russos fogem da regi o de Kharkiv,
            Ucr nia retoma uma grande  rea .
        1666612800: [], # Outubro - Russia come a a alvejar a energia
           el trica da Ucr ncia com ataques a reos. M ssies em Kyev.
            Explos o na ponte Crimea.
        1669291200: [], # Novembro - Kherson    liberada depois de oito
           meses. Blackout massivo.
        1671883200: [], # Dezembro - Russia se torna o pa s mais
           sancionado. 100000 soldados russos perdidos. Zelensky visita
           USA.
        1674561600: [], # Janeiro - Alemanha concorda em enviar tanques.
        1677240000: [], # Fevereiro - Biden visita Kyev.
        1679670000: [], # Mar o
        1682348400: [], # Abril
        1684940400: [], # Maio
        1687618800: [] # Junho
    }

    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToCountUsers(data, dateDict, dateDictCopy,
           checkedUsersDict)

    return dateDict, dateDictCopy

def countPositiveOrNegative(file, positiveWordCountDict,
   negativeWordCountDict):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToCountPositiveOrNegative(data, positiveWordCountDict,
           negativeWordCountDict)
```

```
# Gera o gr fico de volume de tweets e retweets durante o per odo
    coletado.
def generateVolumeGraph(
        targetDict,
        title: str = '',
        xlabel: str = '',
        ylabel: str = ''
) -> None:

    selectedLabel = 'Replies'
    selectedColor = 'red'
    selectedDict = targetDict

    plt.plot(list(selectedDict.keys()), list(selectedDict.values()),
        label=selectedLabel, color=selectedColor)

    ymin, ymax = plt.axis()[2:4]
    ylen = ymax - ymin
    space_label_dot = (ylen * 2) / 100

    point_labels = [str(i + 1) for i in range(0, len(list(selectedDict.
        keys())) + 1)]

    # Adicionar pontos na linha 'Tweets'.
    for date in list(selectedDict.keys()):
        if date in list(selectedDict.keys()):
            plt.scatter(date, selectedDict[date], marker='o', color=
                selectedColor)
            plt.text(date, selectedDict[date] +
                    space_label_dot, point_labels.pop(0))

    # Adicionar legenda
    plt.legend()

    # Adicionar t tulos e r tulos
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)


    # PARA USO NA CONTAGEM DE POSITIVAS E NEGATIVAS
    plt.xticks(list(selectedDict.keys()), rotation='vertical')

    # Mostrar o gr fico
    plt.show()
```

```
def generateVolumeGraphAllSubs(
        targetDict,
        colorList,
        count,
        label,
        title: str = '',
        xlabel: str = '',
        ylabel: str = ''
) -> None:

    selectedColor = colorList[count]
    selectedDict = targetDict

    plt.plot(list(selectedDict.keys()), list(selectedDict.values()),
       color=colorList[count], label=label)

    ymin, ymax = plt.axis()[2:4]
    ylen = ymax - ymin
    space_label_dot = (ylen * 2) / 100

    #point_labels = [str(i + 1) for i in range(0, len(list(selectedDict.
       keys())) + 1)]

    # Adicionar pontos na linha 'Tweets'.
    for date in list(selectedDict.keys()):
        if date in list(selectedDict.keys()):
            plt.scatter(date, selectedDict[date], marker='o', color=
               selectedColor)
            # plt.text(date, selectedDict[date] +
            #          space_label_dot, point_labels.pop(0))

    plt.legend(loc='upper left', bbox_to_anchor=(1, 1))

    # PARA USO NA CONTAGEM DE POSITIVAS E NEGATIVAS
    plt.xticks(list(selectedDict.keys()), rotation='vertical')

def generatePositiveXNegativeCountGraph(positiveWordCountDict,
   negativeWordCountDict, title='',xlabel='',ylabel=''):

    positiveLabelColor = 'blue'
    positiveLabel = 'Positive'

    negativeLabelColor = 'red'
    negativeLabel = 'Negative'

    plt.plot(list(positiveWordCountDict.keys()), list(
       positiveWordCountDict.values()), label=positiveLabel, color=
```

```
        positiveLabelColor)
    plt.plot(list(negativeWordCountDict.keys()), list(
        negativeWordCountDict.values()), label=negativeLabel, color=
        negativeLabelColor)

    ymin, ymax = plt.axis()[2:4]
    ylen = ymax - ymin
    space_label_dot = (ylen * 2) / 100

    point_labels = [str(i + 1) for i in range(0, len(list(
        positiveWordCountDict.keys())) + 1)]
    point_labels2 = [str(i + 1) for i in range(0, len(list(
        negativeWordCountDict.keys())) + 1)]

    for date in list(positiveWordCountDict.keys()):
        if date in list(positiveWordCountDict.keys()):
            plt.scatter(date, positiveWordCountDict[date], marker='o',
                color=positiveLabelColor)
            plt.text(date, positiveWordCountDict[date] +
                    space_label_dot, point_labels.pop(0))

    for date in list(negativeWordCountDict.keys()):
        if date in list(negativeWordCountDict.keys()):
            plt.scatter(date, negativeWordCountDict[date], marker='o',
                color=negativeLabelColor)
            plt.text(date, negativeWordCountDict[date] +
                    space_label_dot, point_labels2.pop(0))

    plt.legend()
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.xticks(list(positiveWordCountDict.keys()), rotation='vertical')
    plt.show()

# Gera a nuvem de palavras mais populares (top-100) dos tweets e
    retweets.
def generateWordCloud(wordCloudText, generateWithBigrams) -> None:
    # Remove caracteres especiais do texto.
    text = re.sub(r'[^a-zA-Z0-9\s]', '', wordCloudText)

    # Divide a string em palavras.
    words = nltk.word_tokenize(text)

    # Transforma todas as palavras em min sculas.
    words_lower = [str(word).lower() for word in words]
```

```
    # Remove as stopwords.
    words_filtered = [lemmatizer.lemmatize(word) for word in words_lower
        if word.isalnum() and word not in stop_words]

    if generateWithBigrams:
        bi_grams = list(bigrams(words_filtered))
        bi_grams_filtered = [tupla[0] + ' ' + tupla[1] for tupla in
           bi_grams]
        word_freq = Counter(bi_grams_filtered)
        top_words = word_freq.most_common(50)
        lista_bigrams = [tupla[0] for tupla in top_words]
        lista_bigrams.sort(key=len)
        print(lista_bigrams)
    else:
        word_freq = Counter(words_filtered)
        top_words = word_freq.most_common(50)
        lista_str = [tupla[0] for tupla in top_words]
        lista_str = list(set(lista_str))
        lista_str.sort(key=len)
        print(lista_str)

    # Cria a nuvem de palavras.
    wordcloud = WordCloud(
        width=800,
        height=400,
        background_color='white'
    ).generate_from_frequencies(dict(top_words))
    plt.figure(figsize=(12, 10))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.show()


# Conta o n mero de tweets/retweets por palavra-chave.
def count_posts_or_comments_per_keyword(dataset: pd.DataFrame):
    num_posts_per_keyword = {}

    for keyword in keywordList:
        for row in dataset['Title']:
            if keyword.lower() in row.lower():
                if keyword.lower() in list(num_posts_per_keyword.keys())
                    :
                    num_posts_per_keyword[keyword.lower()] += 1
                else:
                    num_posts_per_keyword[keyword.lower()] = 1
```

```
def countAppearancesForAllKeywords(file, keywordList, targetDict) ->
    dict:
    for keyword in keywordList:
        countKeywordAppearancesByDate(file, keyword.lower(), targetDict)

def countSentimentPolarityForAllKeywords(file, keywordList, targetDict):
    for keyword in keywordList:
        countSentimentPolarityByKeywordDate(file, keyword.lower(),
            targetDict)

def getSentimentPolarityMean(globalDictReplies):
    for date in globalDictReplies:
        globalDictReplies[date] = (
            globalDictReplies[date]['Total Polarity']/
            globalDictReplies[date]['Comment Count'])

def getSentimentPolarityKeywoardMean(targetDict):
    for date in targetDict:
        for keyword in targetDict[date]:
            if targetDict[date][keyword] is not np.nan:
                targetDict[date][keyword] = (
                    targetDict[date][keyword]['keywordPolarityTotal']/
                    targetDict[date][keyword]['keywordCount'])

def deleteKeysFromDict(targetDict):
    listToDelete = []
    for date in targetDict:
        for keyword in targetDict[date]:
            if targetDict[date][keyword] == 0:
                listToDelete.append(targetDict[date][keyword])

    for item in listToDelete:
        del item

# Gera gr fico heatmap de tweets ou retweets por semanas e palavras-
    chave.
def generateHeatmap(
        data_tweets_per_week_and_keyword: dict,
        cmap: str = 'RdYlGn',
        title: str = '',
        x_label: str = '',
        y_label: str = '',
        label_color_bar: str = '',
        log_scale: bool = False
) -> None:
    keywords = list(data_tweets_per_week_and_keyword[list(
        data_tweets_per_week_and_keyword.keys())[0]].keys())
```

```
    weeks = list(data_tweets_per_week_and_keyword.keys())
    data_tweets_per_keyword = [list(week.values()) for week in
        data_tweets_per_week_and_keyword.values()]
    data_tweets_per_keyword_inverted = [list(tupla) for tupla in zip(*
        data_tweets_per_keyword)]

    # Crie um array de dados.
    dados = np.array(data_tweets_per_keyword_inverted)


    # Crie o gr fico de heatmap usando a fun   o imshow do Matplotlib.
    fig, ax = plt.subplots()
    if log_scale:
        heatmap = ax.imshow(dados, cmap=cmap, interpolation=None, norm=
            LogNorm())
    else:
        heatmap = ax.imshow(dados, cmap=cmap, interpolation=None)

    # Adicione a barra de cores.
    cax = ax.figure.colorbar(heatmap, ax=ax)

    # Defina o texto da legenda.
    cax.set_label(label_color_bar)

    # Adicione as etiquetas dos eixos.
    ax.set_xticks(np.arange(dados.shape[1]))
    ax.set_yticks(np.arange(dados.shape[0]))
    ax.set_xticklabels(weeks)
    ax.set_yticklabels(keywords)

    # Rotacione os r tulos do eixo x.
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
        rotation_mode="anchor")

    # Adicione o t tulo.
    ax.set_title(title)

    # Adicione a label do eixo x.
    ax.set_xlabel(x_label)

    # Adicione a label do eixo y.
    ax.set_ylabel(y_label)

    # Mostre o gr fico.
    plt.show()
```

```
def generate_liwc_analysis_of_datasets_by_keyword(keywords_dir: str) ->
    dict:
    liwc_analysis = {}
    for keyword in os.listdir(keywords_dir):
        folder_path = os.path.join(keywords_dir, keyword)
        if os.path.isdir(folder_path):
            csv_files = [arquivo for arquivo in os.listdir(
                folder_path) if arquivo.endswith('.csv')]
            if csv_files:
                datasets = []
                for csv_file in csv_files:
                    csv_file_path = os.path.join(folder_path, csv_file)
                    dataset = pd.read_csv(csv_file_path)
                    datasets.append(dataset)
                dataset_keyword = pd.concat(datasets)
                liwc_dataset_keyword = liwc.process_df(dataset_keyword,
                    'Content')
                liwc_analysis[keyword] = liwc_dataset_keyword
    return liwc_analysis

def apply_kruscal_test_in_liwc_analysis(liwc_analysis_list: list) ->
    list:
    liwc_analysis_list = [liwc_analysis.fillna(0) for liwc_analysis in
        liwc_analysis_list]
    selected_attributes = []
    for i in range(len(liwc_analysis_list)):
        for j in range(len(liwc_analysis_list)):
            if i == j:
                continue
            for attribute in liwc_analysis_list[i].columns:
                if attribute in liwc_analysis_list[j].columns:
                    _, p_value = kruskal(
                        liwc_analysis_list[i][attribute],
                            liwc_analysis_list[j][attribute])
                    significance_level = 0.05
                    if p_value < significance_level:
                        selected_attributes.append(attribute)
    selected_attributes = list(set(selected_attributes))
    return selected_attributes

def apply_gini_coefficient_to_pick_the_most_discriminating_attributes(
        liwc_analysis_per_keyword: dict,
        selected_attributes: list = None
) -> pd.DataFrame:
    scores_keyword =
        generate_table_with_average_analysis_liwc_by_keyword(
        liwc_analysis_per_keyword)
```

```
    if not selected_attributes:
        selected_attributes = scores_keyword.columns
    scores_keyword = scores_keyword.loc[:, selected_attributes]
    scores_keyword = scores_keyword.transpose()

    baseline = scores_keyword.mean(axis=1)
    scores_rel = scores_keyword.divide(baseline, axis=0)
    whitelist = scores_rel.apply(lambda s: gini(s.values), axis=1).
       sort_values(ascending=False).head(20).index
    scores_rel = scores_rel[scores_rel.index.isin(whitelist)]
    scores_rel.index = scores_rel.index.map(lambda s: s.title())
    scores_rel.rename(index=attribute_translation, inplace=True)

    scores_rel_zscore = (scores_rel - scores_rel.mean()) / scores_rel.
       std()
    return scores_rel_zscore

def gini(x: list) -> float:
    total = 0
    for i, xi in enumerate(x[:-1], 1):
        total += np.sum(np.abs(xi - x[i:]))
    return total / (len(x)**2 * np.mean(x))

def generate_table_with_average_analysis_liwc_by_keyword(
   liwc_analysis_per_keyword: dict) -> pd.DataFrame:
    liwc_analysis_per_keyword_mean = {}
    for keyword, liwc_analysis in liwc_analysis_per_keyword.items():
        liwc_analysis_mean = liwc_analysis.mean(axis=0).to_frame()
        liwc_analysis_per_keyword_mean[keyword] = liwc_analysis_mean.
           transpose(
        )

    table = pd.DataFrame(
        pd.concat(list(liwc_analysis_per_keyword_mean.values())))
    table.insert(0, '', list(liwc_analysis_per_keyword_mean.keys()))
    table.set_index('', inplace=True)
    return table


def generateCooccurrenceNetwork(filePath):
    dataset = pd.read_csv(filePath)
    cleanedDataset = utils.remove_special_characters(dataset.to_string()
       )
    dataset_1 = pd.read_csv('data/csvSubsContentOnly/EndlessWar.csv')
    dataset_2 = pd.read_csv('data/csvSubsContentOnly/News.csv')
    dataset_3 = pd.read_csv('data/csvSubsContentOnly/Politics.csv')
```

```
dataset_5 = pd.read_csv('data/csvSubsContentOnly/
    RussiaUkraineWar2022.csv')
dataset_6 = pd.read_csv('data/csvSubsContentOnly/Ukraine.csv')
dataset_7 = pd.read_csv('data/csvSubsContentOnly/
    UkraineWarVideoReport.csv')
dataset_8 = pd.read_csv('data/csvSubsContentOnly/WorldNews.csv')

dfs = [
    dataset_1,
    dataset_2,
    dataset_3,
    dataset_5,
    dataset_6,
    dataset_7,
    dataset_8
]

concatenated_df = pd.concat(dfs, ignore_index=True)

occurrences = []

df = pd.read_csv(filePath)
co_occurrence_matrix = {}
for index, row in concatenated_df.iterrows():
    for k1, k2 in itertools.combinations(Cooccurrencekeywords, 2):
        if k1.lower() in str(row).lower() and k2.lower() in str(row)
            .lower():
            try:
                co_occurrence_matrix[(k1.lower(), k2.lower())] += 1
            except Exception:
                co_occurrence_matrix[(k1.lower(), k2.lower())] = 1

print(co_occurrence_matrix)


# Step 4: Generate Co-occurrence Network using NetworkX
G = nx.Graph()

for pair, count in co_occurrence_matrix.items():
    G.add_edge(pair[0], pair[1], weight=count)

node_sizes = {node: G.degree(node) * 200 for node in G.nodes}
edge_widths = [data['weight'] for _, _, data in G.edges(data=True)]

min_normalized_weight = 0.5  # Example minimum value
max_normalized_weight = 5  # Example maximum value
```

```
    min_edge_weight = min(edge_widths)
    max_edge_weight = max(edge_widths)

    normalized_edge_weights = [
        min_normalized_weight + (max_normalized_weight -
            min_normalized_weight) * ((w - min_edge_weight) / (
            max_edge_weight - min_edge_weight))
        for w in edge_widths
    ]

    pos = nx.spring_layout(G, k=0.1, iterations=300)  # Positions for
        all nodes
    nx.draw_networkx_nodes(G, pos, node_color='lightblue', node_size=
        list(node_sizes.values()))
    nx.draw_networkx_edges(G, pos, edgelist=G.edges(), width=
        normalized_edge_weights, alpha=0.5, edge_color='gray')
    nx.draw_networkx_labels(G, pos, font_size=12)

    plt.show()

def count_occurrences(tokens):
    return {token: tokens.count(token) for token in set(tokens)}

def generateThreadGraph(file):
    G = nx.DiGraph()

    with open(f'./data/jsonSubs/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToGenerateThreadGraph(data, G)

    nx.write_gexf(G, "graph.gexf")

def generateUserConnectionsGraph(file):
    G = nx.DiGraph()

    logging.info('Come ou a iterar.')
    with open(f'./data/jsonSubs/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToGenerateUserConnectionGraph(data, G)

    logging.info('Terminou de iterar.')

    nx.write_gexf(G, f'./data/gephiFiles/{file[:-5]}.gexf')

def get_communities_by_louvain(file: str, is_directed: bool, out_dir:
    str) -> None:
```

```
    graph = nx.read_gexf(f'./data/gephiFilesFiltered/{file}')

    # Crie um grafo n o direcionado, se necess rio.
    if is_directed:
        graph = graph.to_undirected()

    # Execute o algoritmo Louvain para detec  o de comunidades.
    coms = cd_lib_algos.louvain(graph)
    mod = coms.newman_girvan_modularity()
    info_dict = {'name': f'{file[:-5]}', 'modularity': mod, 'number of
        communities': len(coms.communities)}

    if not (os.path.exists(out_dir)):
        os.mkdir(out_dir)
    if not (os.path.exists(f'{out_dir}/louvain')):
        subfolder = f'{out_dir}/louvain/'
        os.mkdir(subfolder)
    if not (os.path.exists(f'{out_dir}/louvain/{file[:-5]}')):
        subfolder = f'{out_dir}/louvain/{file[:-5]}'
        os.mkdir(subfolder)

    count = 1
    for community in coms.communities:
        subgraph = graph.subgraph(community)
        for node in subgraph.nodes():
            subgraph.nodes[node]['community'] = f'com {count}'

        nx.write_gexf(subgraph, out_dir + '/' + 'louvain' + '/' + f'{
            file[:-5]}' + '/' + f'{file[:-5]}_{count}.gexf')


        if count == 10:
            break
        count += 1

    nx.write_gexf(graph, out_dir + '/' + 'louvain' + '/' + f'{file[:-5]}
        ' + '/' + f'{file[:-5]}_full_graph.gexf')
    with open(out_dir + '/' + 'louvain' + '/' + f'{file[:-5]}' + '/' + f
        '{file[:-5]}_info.txt', 'w') as f:
        json.dump(info_dict, f)

def get_communities_by_leiden(file: str, is_directed: bool, out_dir: str
    ) -> None:

    graph = nx.read_gexf(f'./data/gephiFilesFiltered/{file}')

    # Crie um grafo n o direcionado, se necess rio.
```

```
    if is_directed:
        graph = graph.to_undirected()


    # Execute o algoritmo Louvain para detec  o de comunidades.
    coms = cd_lib_algos.leiden(graph)
    mod = coms.newman_girvan_modularity()
    info_dict = {'name': f'{file[:-5]}', 'modularity': mod, 'number of
       communities': len(coms.communities)}

    if not (os.path.exists(out_dir)):
        os.mkdir(out_dir)
    if not (os.path.exists(f'{out_dir}/leiden')):
        subfolder = f'{out_dir}/leiden/'
        os.mkdir(subfolder)
    if not (os.path.exists(f'{out_dir}/leiden/{file[:-5]}')):
        subfolder = f'{out_dir}/leiden/{file[:-5]}'
        os.mkdir(subfolder)

    count = 1
    for community in coms.communities:
        subgraph = graph.subgraph(community)
        for node in subgraph.nodes():
            subgraph.nodes[node]['community'] = f'com {count}'

        nx.write_gexf(subgraph, out_dir + '/' + 'leiden' + '/' + f'{file
           [:-5]}' + '/' + f'{file[:-5]}_{count}.gexf')


        if count == 10:
            break
        count += 1

    nx.write_gexf(graph, out_dir + '/' + 'leiden' + '/' + f'{file[:-5]}'
        + '/' + f'{file[:-5]}_full_graph.gexf')
    with open(out_dir + '/' + 'leiden' + '/' + f'{file[:-5]}' + '/' + f'
       {file[:-5]}_info.txt', 'w') as f:
        json.dump(info_dict, f)

def get_communities_by_girvan_newman(file: str, is_directed: bool,
   out_dir: str) -> None:

    graph = nx.read_gexf(f'./data/gephiFiles/{file}')

    # Crie um grafo n o direcionado, se necess rio.
    if is_directed:
        graph = graph.to_undirected()
```

```
    # Execute o algoritmo Louvain para detec  o de comunidades.
    coms = nx.community.girvan_newman(graph)
    for communities in itertools.islice(coms, 10):
        print(tuple(sorted(c) for c in communities))

    num_communities = sum(1 for _ in next(coms))
    best_partition = max(coms, key=lambda x: nx.community.quality.
       modularity(graph, x))
    mod = nx.community.quality.modularity(graph, best_partition)
    info_dict = {'name': f'{file[:-5]}', 'modularity': mod, 'number of
       communities': num_communities}
    print(info_dict)


def get_top_nodes(gexf_file, json_file, result_path):

    graph = nx.read_gexf(gexf_file)

    if not (os.path.exists(result_path)):
        os.mkdir(result_path)

    node_degrees = dict(graph.degree())

    # Step 3: Sort nodes by degree in descending order
    sorted_nodes = sorted(node_degrees, key=node_degrees.get, reverse=
       True)

    # Step 4: Get top 5 nodes with highest degrees
    top_3_nodes = sorted_nodes[:3]

    with open(json_file, 'r') as json_file:
        json_data = json.load(json_file)

    # Print the top 5 nodes and their degrees
    for node in top_3_nodes:
        comments_list = []

        utils.iterate_to_get_comments(json_data, comments_list, node)

        sorted_comment_list = sorted(comments_list, key=lambda x: x['
           Score'], reverse=True)

        # Get the top 5 dictionaries with the highest scores
        top_10_comments = sorted_comment_list[:10]
```

```
        with open(f"{result_path}/{node}.json", "w") as outfile:
            json.dump(top_10_comments, outfile, indent=4)

def get_all_nodes(gexf_file, json_file, result_path):

    graph = nx.read_gexf(gexf_file)

    if not (os.path.exists(result_path)):
        os.mkdir(result_path)

    node_degrees = dict(graph.degree())

    # Step 3: Sort nodes by degree in descending order
    sorted_nodes = sorted(node_degrees, key=node_degrees.get, reverse=
       True)

    with open(json_file, 'r') as json_file:
        json_data = json.load(json_file)

    # Print the top 5 nodes and their degrees
    all_comment_list = []
    for node in sorted_nodes:
        comments_list = []

        utils.iterate_to_get_comments(json_data, comments_list, node)

        sorted_comment_list = sorted(comments_list, key=lambda x: x['
           Score'], reverse=True)
        for comment in sorted_comment_list:
            all_comment_list.append(comment)

    with open(f"{result_path}/all_comments.json", "w") as outfile:
        json.dump(all_comment_list, outfile, indent=4)
```

Listing A.1 – Tratamento de dados do Reddit

```
import praw
import json
import csv
import codecs
import time
import random
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import Select
```

```
from selenium.common.exceptions import NoSuchElementException

PATH = 'C:\Windows\chromedriver.exe'
driver = webdriver.Chrome(PATH)

reddit_read_only = praw.Reddit(
    client_id="6Ewi9quPUyI-yMwk4VayLw",
    client_secret="0416jp3KOAZeAUGZ9bKEHOGRnX2j_g",
    user_agent="EVB scraper")

def process_replies(comment_list):
    reply_list = []
    comment_list.replace_more(limit=0)
    for comment in comment_list:
        comment_dict = {}
        if comment.author == None:
            comment_dict["Author"] = ""
        else:
            comment_dict["Author"] = comment.author.name
        comment_dict["ID"] = comment.id
        comment_dict["Comment Date"] = comment.created_utc
        comment_dict["Comment Score"] = comment.score
        comment_dict["Comment Content"] = comment.body
        comment_dict["Number of Replies"] = len(comment.replies.list())
        comment_dict["Comment Replies"] = process_replies(comment.
           replies)
        reply_list.append(comment_dict)

    return reply_list

def iterate_through_results():
    driver.implicitly_wait(10)
    search_results = driver.find_elements(By.CSS_SELECTOR, "div.g")
    print(len(search_results))
    counter = 0
    submission_list = []
    for result in search_results:
        link = result.find_element(By.CSS_SELECTOR, "a")
        try:
            submission = reddit_read_only.submission(
            url=link.get_attribute('href'))
        except praw.exceptions.InvalidURL:
            continue
        submission_dict = {}
        submission_dict["Title"] = submission.title
        if submission.author == None:
            submission_dict["Author"] = ""
```

```
        else:
            submission_dict["Author"] = submission.author.name
        submission_dict["ID"] = submission.id
        submission_dict["Url"] = submission.url
        submission_dict["Time"] = submission.created_utc
        submission_dict["Score"] = submission.score
        submission_dict["Submission Content"] = submission.selftext
        submission_dict["Number of Comments"] = submission.num_comments
        submission.comments.replace_more(limit=0)
        comment_list = process_replies(submission.comments)
        submission_dict['Comment List'] = comment_list
        submission_list.append(submission_dict)
        counter += 1
        time.sleep(random.randint(10,15))
    return submission_list

final_list = []
driver.get('https://www.google.com')
start_month = '02'
end_month = '03'
for i in range(10):
    search = driver.find_element(By.NAME, 'q')
    search.clear()
    #search parameters
    term_search = 'ukraine'
    site_search = 'site:reddit.com/r/endlesswar'
    time_search_before = f'before:2022-{end_month}-20'
    time_search_after = f'after:2022-{start_month}-20'
    search.send_keys(
        f'{term_search}' +
        f' {site_search}' +
        f' {time_search_before}' +
        f' {time_search_after}')
    search.send_keys(Keys.RETURN)
    submission_list = []
    for i in range(30):
        print(f'{i} page')
        submission_list += iterate_through_results()
        try:
            next_button = driver.find_element(By.ID, "pnnext")
            next_button.click()
        except NoSuchElementException:
            print("breaking")
            break
    final_list += submission_list
    start_month = '0' + str(int(start_month) + 1)
    end_month = '0' + str(int(end_month) + 1)
```

```
file_path = 'EndlessWarTimeline.json'

with open(file_path, 'w') as json_file:
    json.dump(final_list, json_file, indent=4)
```

Listing A.2 – Extração de dados

```
import json
import csv
from typing import Any
import pandas as pd
from collections import Counter
from datetime import datetime
import zipfile
import os.path
from global_vars import *
from vaderSentiment import vaderSentiment as vs
import re

sentiment_analyzer = vs.SentimentIntensityAnalyzer()

def addPolarityColumnToJsons(file):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)

    iterateToAddSentimentColumn(data)

    with open(f'SP{file}', 'w') as json_file:
        json.dump(data, json_file, indent=4)

def addSentimentLabelToJsons(file):
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)

    iterateToAddSentimentLabelColumn(data)

    with open(f'SPLabel{file}', 'w') as json_file:
        json.dump(data, json_file, indent=4)

def iterateToAddSentimentColumn(dic):
    for data in dic:
        try:
            data['Polarity Score'] = sentiment_analyzer.polarity_scores(
                data['Submission Content'])['compound']
            iterateToAddSentimentColumn(data['Comment List'])
        except Exception:
```

```
            data['Polarity Score'] = sentiment_analyzer.polarity_scores(
                data['Comment Content'])['compound']
            iterateToAddSentimentColumn(data['Comment Replies'])

def iterateToAddSentimentLabelColumn(dic):
    for data in dic:
        try:
            polarityScore = data['Polarity Score']
            if polarityScore > 0:
                sentimentLabel = 'Positive'
            else:
                sentimentLabel = 'Negative'
            data['Sentiment Label'] = sentimentLabel
            iterateToAddSentimentLabelColumn(data['Comment List'])
        except Exception:
            polarityScore = data['Polarity Score']
            if polarityScore > 0:
                sentimentLabel = 'Positive'
            else:
                sentimentLabel = 'Negative'
            data['Sentiment Label'] = sentimentLabel
            iterateToAddSentimentLabelColumn(data['Comment Replies'])

def merge_json_files(file_paths):
    merged_contents = []

    for file_path in file_paths:
        with open(file_path, 'r', encoding='utf-8') as file_in:
            merged_contents.extend(json.load(file_in))

    with open('WorldNewsMerged.json', 'w', encoding='utf-8') as file_out
        :
        json.dump(merged_contents, file_out, indent=4)

def remove_duplicates(file_path):

    with open(file_path, 'r', encoding='utf-8') as file_in:
        id_list = []
        unique = []
        no_dups = json.load(file_in)
        for dic in no_dups:
            if dic['ID'] not in id_list:
                id_list.append(dic['ID'])
                unique.append(dic)

    with open('WorldNewsNoDuplicates.json', 'w', encoding='utf-8') as
        file_out:
```

```
        json.dump(unique, file_out, indent=4)

def json_to_csv_converter():
    path = "./data"
    dir = os.listdir(path)

    for file in dir:
        if 'NegativeSPLabeledWorldNewsFiltered' in file:
            convertedFile = pd.read_json('./data/' + file)
            convertedFile.to_csv(f'./data/ + {file}.csv')

def convertJsonContentToCsv():
    path = './data/jsonSubsContentOnly'
    for file in os.listdir(path):
        convertedFile = pd.read_json(path + '/' + file)
        convertedFile.to_csv(f'data/csvSubsContentOnly/{file[:-5]}.csv')

def convertJsonKeywordsToCsv():
    path = './data/keywords'
    keywordsDir = os.listdir(path)

    for keywordDir in keywordsDir:
        keywordDirPath = f'./data/keywords/{keywordDir}'
        jsonKeywordFiles = os.listdir(keywordDirPath)
        for jsonKeywordFile in jsonKeywordFiles:
            convertedFile = pd.read_json(keywordDirPath + '/' +
                jsonKeywordFile)
            convertedFile.to_csv(f'{keywordDirPath}/{jsonKeywordFile
                [:-5]}.csv')

def filterKeywords(file_path):
    keywords = 'Putin Russia Ukraine War Russian Ukrainian Conflict
        Vladimir Zelensky'

    with open(file_path, 'r', encoding='utf-8') as file_in:
        filteredDic = json.load(file_in)
        for dic in filteredDic:
            validDic = False
            for keyword in keywords.split():
                if keyword in dic['Title']:
                    validDic = True
            if validDic == False:
                filteredDic.remove(dic)

    with open('NewsFiltered.json', 'w', encoding='utf-8') as file_out:
        json.dump(filteredDic, file_out, indent=4)
```

```
def list_sorter(listToSort):
    sorted_list = sorted(listToSort, key=lambda x: len(x))

    for element in sorted_list:
        print(f"'{element}',")

def iterateToFilterSentimentValue(data, newDataList, sentimentValue):
    for content in data:
        try:
            if content['Sentiment Label'] == sentimentValue:
                if content['Submission Content'] == '':
                    newDataList.append({'Content': 'none'})
                else:
                    newDataList.append({'Content': content['Submission
                        Content']})
            iterateToFilterSentimentValue(content['Comment List'],
                newDataList, sentimentValue)
        except Exception:
            if content['Sentiment Label'] == sentimentValue:
                if content['Comment Content'] == '':
                    newDataList.append({'Content': 'none'})
                else:
                    newDataList.append({'Content': content['Comment
                        Content']})
            iterateToFilterSentimentValue(content['Comment Replies'],
                newDataList, sentimentValue)

def filterOnlyPositiveOrNegativeKeys(file, sentimentValue):
    newDataList = []
    with open(f'./data/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToFilterSentimentValue(data, newDataList, sentimentValue)

    with open(sentimentValue + file, 'w', encoding='utf-8') as file_out:
        json.dump(newDataList, file_out, indent=4)

def iterateToFilterKeyword(data, keyword, newList):
    for content in data:
        try:
            if keyword in content['Submission Content'].lower():
                newList.append({'Content': content['Submission Content'
                    ]})
            iterateToFilterKeyword(content['Comment List'], keyword,
                newList)
        except Exception:
            if keyword in content['Comment Content'].lower():
                newList.append({'Content': content['Comment Content']})
```

```
            iterateToFilterKeyword(content['Comment Replies'], keyword,
                newList)

def iterateToFilterContent(data, newList):
    for content in data:
        try:
            newList.append({'Content': content['Submission Content']})
            iterateToFilterContent(content['Comment List'], newList)
        except Exception:
            newList.append({'Content': content['Comment Content']})
            iterateToFilterContent(content['Comment Replies'], newList)

def iterateToIncludeParentId(data):
    for userInteraction in data:
        try:
            for userInteractionChild in userInteraction['Comment List']:
                userInteractionChild['Parent ID'] = userInteraction['ID'
                    ]
                iterateToIncludeParentId(userInteraction['Comment List'
                    ])

        except Exception:
            for userInteractionChild in userInteraction['Comment Replies
                ']:
                userInteractionChild['Parent ID'] = userInteraction['ID'
                    ]
                iterateToIncludeParentId(userInteraction['Comment
                    Replies'])

def filterFilesByKeyword():
    for keyword in keywordList:
        for file in os.listdir(f'./data/jsonSubs/'):
            newList = []
            with open(f'./data/jsonSubs/{file}', 'r') as json_file:
                print(json_file)
                data = json.load(json_file)
                iterateToFilterKeyword(data, keyword.lower(), newList)

            with open(f'./data/keywords/{keyword}/{keyword}{file}', 'w',
                encoding='utf-8') as file_out:
                json.dump(newList, file_out, indent=4)

def remove_special_characters(inputString):
    # Define a regular expression pattern to match special characters
    pattern = r'[^a-zA-Z0-9\s]'  # This pattern will keep only
        alphanumeric characters and spaces
```

```
    # Use re.sub() to replace the matched pattern with an empty string
    clean_string = re.sub(pattern, '', inputString)

    return clean_string

def removeSelectedWords(inputString):
    unwantedWords = ['Title', 'Author', 'ID', 'Url', 'Time', 'Score', '
       Submission', 'Content', 'Number', 'of', 'Comments', 'Comment', '
       List','Date', 'Replies']
    for unwantedWord in unwantedWords:
        if unwantedWord.lower() in inputString.lower():
            inputString.replace(unwantedWord, '')

def filterOnlyContentFromCsv():
    for file in os.listdir(f'./data/jsonSubs/'):
            newList = []
            with open(f'./data/jsonSubs/{file}', 'r') as json_file:
                print(json_file)
                data = json.load(json_file)
                iterateToFilterContent(data, newList)

            with open(f'./data/csvSubsContentOnly/{file}', 'w', encoding
               ='utf-8') as file_out:
                json.dump(newList, file_out, indent=4)

def includeParentIDInJsons(file):
    with open(f'./data/jsonSubs/{file}', 'r') as json_file:
        data = json.load(json_file)
        iterateToIncludeParentId(data)

    with open(f'./data/jsonSubsParentKey/{file}', 'w') as json_file:
        json.dump(data, json_file, indent=4)


def remove_lines_with_words(input_file, output_file, words):
    with open(input_file, 'r') as infile, open(output_file, 'w') as
       outfile:
        for line in infile:
            if not any(word in line for word in words):
                outfile.write(line)

def rename_file(directory, str_to_include):
    for filename in os.listdir(directory):
        # Check if the entry is a file
        if os.path.isfile(os.path.join(directory, filename)):
            # Construct the new filename with 'lou' at the beginning
            new_filename = os.path.join(directory, f"{str_to_include}_{
```

```
                filename}")
            # Rename the file
            os.rename(os.path.join(directory, filename), new_filename)
            print(f"Renamed {filename} to {new_filename}")

def iterate_to_get_comments(data, comments_list, node):
    for comment in data:
        try:
            if comment['Author'].lower() == node.lower():
                if 'Comment Content' in comment.keys():
                    new_item = {'Author': comment['Author'].lower(), '
                        Content': comment['Comment Content'], 'Score':
                        comment['Comment Score']}
                    repeat = False
                    for list_item in comments_list:
                        if list_item == new_item:
                            repeat = True
                    if repeat == False:
                        comments_list.append(new_item)
                else:
                    new_item = {'Author': comment['Author'].lower(), '
                        Content': comment['Title'], 'Score': comment['
                        Score']}
                    repeat = False
                    for list_item in comments_list:
                        if list_item == new_item:
                            repeat = True
                    if repeat == False:
                        comments_list.append(new_item)
            iterate_to_get_comments(comment['Comment List'],
                comments_list, node)
        except Exception:
            if comment['Author'].lower() == node.lower():
                if 'Comment Content' in comment.keys():
                    new_item = {'Author': comment['Author'].lower(), '
                        Content': comment['Comment Content'], 'Score':
                        comment['Comment Score']}
                    repeat = False
                    for list_item in comments_list:
                        if list_item == new_item:
                            repeat = True
                    if repeat == False:
                        comments_list.append(new_item)
                else:
                    new_item = {'Author': comment['Author'].lower(), '
                        Content': comment['Title'], 'Score': comment['
                        Score']}
```

```
                    repeat = False
                    for list_item in comments_list:
                        if list_item == new_item:
                            repeat = True
                    if repeat == False:
                        comments_list.append(new_item)
            iterate_to_get_comments(comment['Comment Replies'],
                comments_list, node)

def iterate_to_get_community_csv(all_comments_json_path, output_path):
    with open(all_comments_json_path, 'r') as json_file:
        data = json.load(json_file)

    # Extract 'content' from each item and store in a list
    contents = [item['Content'] for item in data]

    # Write the contents to a CSV file
    with open(output_path, 'w', newline='', encoding='utf-8') as
        csv_file:
        writer = csv.writer(csv_file)
        # Write header if needed
        writer.writerow(['content'])
        # Write contents
        writer.writerows([[content] for content in contents])
```

Listing A.3 – Utilitários

APÊNDICE B – ARTIGO SBC

Análise Do Comportamento De Comunidade Sobre Dados Da Guerra Da Ucrânia No Reddit

Eduardo Vinicius Betim[1], **Carina Friedrich Dorneles**[1], **Eric Fernandes de Mello Araújo**[2]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

[2]Departamento de Ciência da Computação – Universidade Federal de Lavras
Lavras – MG – Brazil

edubetimso@gmail.com, dorneles@inf.ufsc.br, eric@ufla.br

Abstract. *This work uses community detection algorithms on data collected from the social network Reddit, gathered from February 2022 to June 2023, to identify similar groups and analyze user behavior. Through data characterization techniques and graph construction, analyses are conducted to reveal patterns and characteristics of the collected data, providing a better understanding of the social interactions occurring on the social network in relation to conflict.*

Resumo. *Este trabalho utiliza algoritmos de detecção de comunidade em dados coletados da rede social Reddit, coletados no período de Fevereiro de 2022 até Junho de 2023, a fim de identificar grupos semelhantes e analisar o comportamento dos usuários. Por meio de técnicas de caracterização de dados e construção de grafos, são realizadas análises capazes de expor padrões e características dos dados coletados, proporcionando um entendimento melhor das interações sociais ocorridas na rede social com relação ao conflito.*

1. Introdução

Com acessibilidade e capacidade de disseminar informação rapidamente, as redes sociais se tornaram uma ferramenta fundamental de observação da cultura humana, das opiniões e posicionamentos de diferentes círculos sociais, e da repercussão gerada por qualquer evento ao redor do mundo. Qualquer usuário, independentemente de status, fama ou influência, pode expor seu ponto de vista sobre determinado tópico para que outros usuários vejam e interajam, por sua vez adicionando as próprias impressões à discussão. Assim como no mundo real, isso promove a criação de relações interpessoais, convergindo e divergindo indivíduos com base em suas interações, para eventualmente levar a criação de círculos sociais dentro do próprio ambiente virtual [Elena-Iulia 2018].

Naturalmente, o compartilhamento de interesses e opiniões similares faz com que usuários convirjam em grupos, que podemos denominar de comunidades. Com o surgimento destes agrupamentos em mente, alguns autores propõem a utilização de algoritmos de detecção de comunidades em redes sociais que são baseados em teoria dos grafos na ciência da computação e em agrupamentos hierárquicos na sociologia [Newman and Girvan 2004]. O objetivo é utilizar técnicas para encontrar e modelar comunidades dentro de um conjunto de dados em formato de grafo, onde os nodos correspondem aos usuários, e as arestas às interações entre eles. Nesse contexto, comunidades

são vértices que compartilham propriedades comuns, podendo assim um mesmo vértice fazer parte de múltiplas comunidades.

Devido à sua popularidade e também ao fato de todo o material compartilhado vir dos próprios usuários, a rede social Reddit se tornou um excelente meio de adquirir opiniões e informações referentes à guerra entre Rússia e Ucrânia, tema de impacto global que consequentemente provocou grande repercussão no meio virtual. Com isso em mente, ela foi escolhida como alvo da extração de dados, para que estes sejam posteriormente analisados utilizando técnicas de caracterização, como também algoritmos de detecção de comunidades.

Espera-se com este trabalho compreender não apenas os tópicos e opiniões que surgiram decorrente da guerra, como também as variáveis que influenciam as discussões dentro da rede social sobre esse tema. Utilizando conceitos de computação social, será possível identificar as condições que influenciam a disseminação de algum determinado tipo de conteúdo dentro de uma comunidade virtual. Serão aplicados algoritmos de detecção de comunidade sobre os dados coletados, para que por fim as redes de opinião presentes possam ser detectadas e analisadas.

1.1. Objetivos

Este projeto tem como meta utilizar os dados coletados do Reddit sobre a guerra da Ucrânia entre os períodos de Fevereiro de 2022 a Julho de 2023 para realizar detecção de comunidades por meio da construção de grafos. Consequentemente, estes grafos servirão de auxílio para compreender melhor as opiniões presentes dentro do conjunto inicial de dados.

Considerando o objetivo geral, consideram-se objetivos específicos:

- Extração de dados do Reddit utilizando técnicas de *web scraping*;
- Caracterização dos dados extraídos, utilizando principalmente a geração de gráficos como mapas de calor, nuvens de palavras, análises quantitativas e análises de sentimento;
- Modelagem de grafos baseados nas árvores de comentários e respostas de postagens dentro dos dados coletados;
- Aplicação de algoritmos de detecção de comunidades sobre os grafos;

2. Trabalhos Relacionados

Este trabalho utilizou alguns conceitos de análise de redes sociais, análise de sentimentos e detecção de comunidades que estão fortemente relacionados com outros trabalhos similares. Esta seção apresentará alguns destes trabalhos, para esclarecer as atividades já desenvolvidas nessa área de pesquisa e pontuar os princípios que influenciaram direta ou indiretamente este trabalho.

2.1. Reddit como ferramenta de análise

Alguns trabalhos utilizam o Reddit como ferramenta de obtenção e análise de dados, devido à sua popularidade e diversidade de usuários e *subreddits*. Para que isso seja possível, é fundamental sumarizar aspectos relevantes do *site*, como no trabalho de [Anderson 2015], que exemplifica o funcionamento do site, juntamente com o perfil dos

usuários e as vantagens e desvantagens de sua utilização como ferramenta de compartilhamento de informação em escala global, fornecendo um sumário útil do *site*.

Um dos aspectos fundamentais para o desenvolvimento deste trabalho é o uso do Reddit para coleta e extração de dados. Nesse sentido, torna-se importante considerar alguns aspectos como ferramentas de extração mais apropriadas, quais atributos serão mais relevantes para as análises, quais tópicos de estudo são mais interessantes e, naturalmente, os aspectos éticos do uso dessa rede como material de pesquisa. Todos esses pontos são objeto de estudo no trabalho de [Proferes et al. 2021], que destaca algumas características que devem ser consideradas por pesquisadores ao trabalhar com essa rede, como a cultura muito variada nos diferentes fóruns do *site*.

Outro perspectiva importante a ser considerada é a pesquisa e análise na área de comunicações e interações sociais. A organização complexa dos círculos sociais do Reddit e o fato de que praticamente todo o conteúdo compartilhado vem dos próprios usuários naturalmente fornece um ambiente propício para a coleta e o estudo de dados. Entretanto, para que o estudo seja possível em primeiro lugar, é de suma importância que o pesquisador entenda a cultura não apenas do *site* como dos *subreddits* individuais dentro dele, como é mencionado no trabalho de [Hintz and Betts 2022].

2.2. A Guerra russo-ucraniana nas redes sociais

Como mencionado anteriormente neste trabalho, as redes sociais são úteis para monitorar a reação dos usuários a eventos de grande impacto mundial. Naturalmente, a guerra entre Rússia e Ucrânia não foi uma exceção, provocando a criação de diversas comunidades dentro do Reddit dedicadas à discussão e compartilhamento de informação sobre o conflito. Além disso, também surgiram diversos trabalhos acadêmicos inspirados no tema.

As redes sociais, devido ao seu alcance e prevalência global, também servem como ferramenta para os próprios países participantes da guerra. Esse aspecto é bastante explorado no trabalho de [Ciuriak 2022], exemplificando algumas ações tomadas pela Rússia, por exemplo, para criar uma névoa virtual e gerar incerteza sobre suas ações durante o conflito. A percepção do público em geral é grandemente influenciada pelo conteúdo compartilhado nas redes, que naturalmente se torna uma preocupação bem grande para países em estado de guerra.

Para um estudo apropriado do impacto nas guerras nas redes sociais, um passo inicial importante logicamente é identificar os locais virtuais de onde é possível extrair informação relevante. Este foi um ponto considerado durante o desenvolvimento deste trabalho, mas grandemente auxiliado por trabalhos como o de [Zhu et al. 2022], que sumariza quais comunidades dentro do Reddit são mais relevantes para a coleta de dados referentes à guerra.

2.3. Análise e detecção de comunidades por meio de grafos

A seção de desenvolvimento deste trabalho utiliza grafos para realizar a detecção de comunidades sobre os dados coletados do Reddit. Este processo é, na verdade, um uso comum da estrutura de grafos, que aparece em diversos outros trabalhos acadêmicos para atingir objetivos similares ao deste. Adicionalmente, compreender a teoria dos grafos, explicada em trabalhos como o de [Fortunato 2010], é de suma importância para que a análise de sua estrutura seja possível.

A utilização de algoritmos de detecção de comunidades em redes sociais também necessita de um estudo mais aprofundado, devido à imensa diversidade de técnicas e algoritmos diferentes, que geram resultados variados em termos de eficiência e conteúdo gerado. Trabalhos como o de [Chunaev 2020] e [Bedi and Sharma 2016] sumarizam diversos algoritmos, permitindo uma visão mais clara de qual deles são apropriados ou não para alcançar os resultados desejados em um trabalho acadêmico que explora a detecção e análise de comunidades.

3. Caracterização de dados

Esta seção apresenta o processo de caracterização de dados, que consiste de diversas análises realizadas quantitativas e semânticas com o intuito de compreender melhor o material que foi extraído da rede social Reddit.

3.1. Coleta de dados

O processo de coleta de dados foi realizado através de *web scraping*, ou seja, extração de conteúdo diretamente de páginas web. Visando esse objetivo, foram criados *scripts* de coleta usando majoritariamente duas ferramentas: a API própria do Reddit[1], disponível para desenvolvedores em geral e que permite extrair conteúdo de praticamente qualquer lugar no site[2], juntamente com o framework Selenium[3], utilizado para automatizar a procura por conteúdo sobre a guerra entre Rússia e Ucrânia.

O conjunto de dados extraído para as análises apresentadas neste trabalho consiste de postagens e comentários coletados de comunidades (*subreddits*) do Reddit referentes ao período da semana de 20 de Fevereiro de 2022, início da invasão russa na Ucrânia, até a semana de 20 de Junho de 2023, data aproximada do início deste projeto de TCC. Foram escolhidos subreddits que continham conteúdos mais relevantes ou direcionados ao contexto da guerra.

Para o caso de comunidades que tinham muitas postagens não relacionadas à guerra, o conteúdo extraído foi filtrado para selecionar apenas material contendo palavras-chave, como *Biden, Trump, Putin, Zelensky, Ukraine, Ukrainian, Russia, Russian, Ukraine War, Ukraine-Russia War, Kyiv, Crimea, Snake Island e Moscow*. Os *subreddits* (fóruns dentro do Reddit) alvo da coleta foram “r/EndlessWar”, “r/News”, “r/Politics”, “r/RussiaUkraineWar2022”, “r/Ukraine”, “r/UkraineWarVideoReport” e “r/WorldNews”.

3.2. Análise quantitativa

A Figura 1 representa a contagem do número total de postagens e comentários presentes em cada comunidade, em cada mês analisado. Quando a linha dos comentários está acima da de postagens em um mesmo período de tempo, pode-se interpretar que cada postagem teve um engajamento grande, dentro do contexto de cada comunidade.

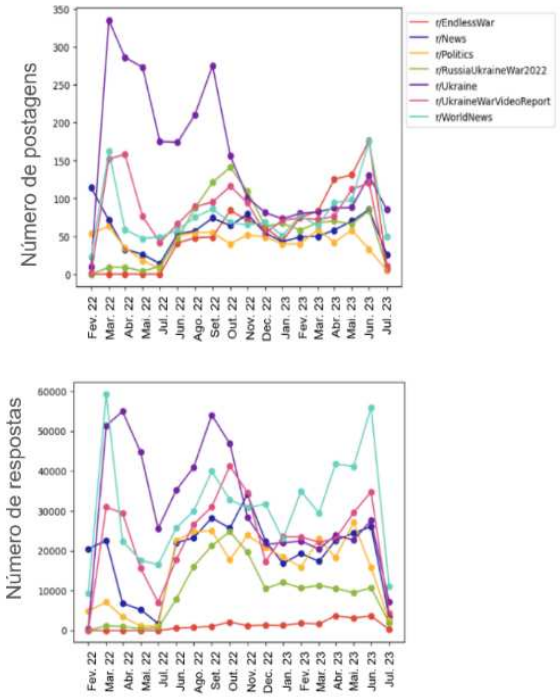
A seguir, na Figura 2, temos as nuvens de palavras, geradas com as 50 palavras mais comuns nas postagens e comentários de cada um dos subreddits analisados.

[1]Disponível em: `https://www.reddit.com/dev/api/`

[2]A API do Reddit sofreu algumas alterações na regulamentação após o período de extração de dados deste trabalho, impondo uma taxa de utilização que antes não existia.

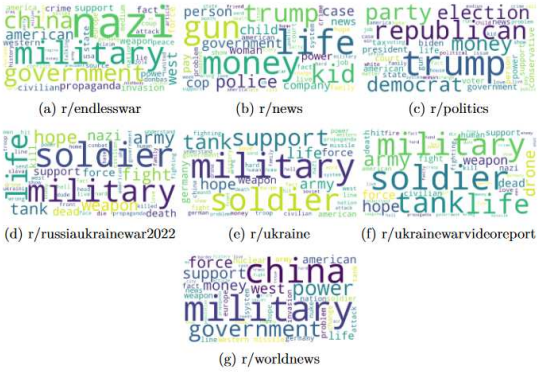
[3]Disponível em: `https://www.selenium.dev/`

Figure 1. Contagem de postagens e respostas por mês



Em geral, grande parte dos termos presentes são termos militares devido à discussão em torno da guerra. Conseguimos perceber também a ocorrência de diversos termos referenciando entidades influentes na guerra, como por exemplo a China ou o ex-presidente americano Donald Trump. É possível que a presença destes termos se deva ao caráter das postagens nessas comunidades, que frequentemente são notícias sobre agentes influenciadores no conflito.

Figure 2. Nuvens de palavra dos termos mais frequentes em cada *subreddit*



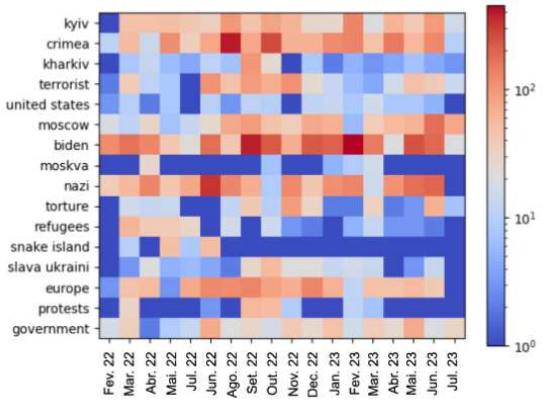
(a) r/endlesswar (b) r/news (c) r/politics

(d) r/russiaukrainewar2022 (e) r/ukraine (f) r/ukrainewarvideoreport

(g) r/worldnews

A Figura 3 fornece um mapa de calor, onde a cor de cada célula é representativa da quantidade de ocorrências por mês nos *subrredits* analisados de algumas palavras-chave relevantes à guerra. Nota-se uma frequência constante do termo nazi, como previamente mostrado pelas nuvens de palavra. A expressão "*slava ukraini*", por sua vez, se mostra presente nos meses de Setembro e Outubro, marcados por um grande contra-ataque da Ucrânia.

Figure 3. Mapa de calor dos termos mais frequentes em cada mês

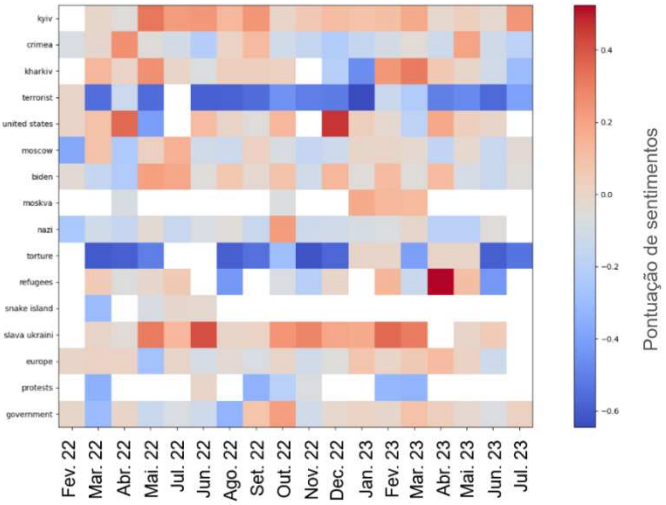


3.3. Análise Semântica

A análise de sentimentos permite compreender melhor a atitude expressada pelos usuários em suas postagens ou comentários. Utilizamos a ferramenta VADER, usando como base o conjunto de dados de todos os *subreddits* analisados neste trabalho, para extrair uma pontuação que representa a conotação sentimental de cada interação dentro da rede social. Conforme descrito pela própria ferramenta, uma pontuação maior que 0,05 representa um sentimento positivo, enquanto uma pontuação menor que -0,05 representa um sentimento negativo.

A Figura 4 mostra um mapa de calor onde cada palavra chave recebe a pontuação de sentimentos dos comentários onde ela aparece, para cada mês de guerra.

Figure 4. Mapa de calor dos termos mais frequentes em cada mês

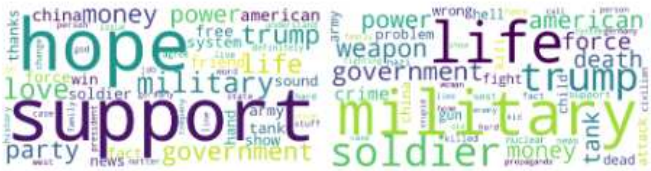


Notamos que as palavras “torture” (tortura) e “terrorist” (terrorista) receberam pontuação considerada negativa em praticamente toda a extensão do gráfico, indicando que, seja qual for o contexto em que apareceram na rede social, geralmente estavam associadas a comentários negativos. A expressão “*slava ukraini*”, por sua vez, teve pontuação muito positiva em determinados momentos e neutra em outros, possivelmente por apare-

cer em comentários de suporte à Ucrânia independentemente de eles possuírem uma conotação mais positiva ou negativa.

A partir da divisão de palavras entre positivas e negativas, baseado na pontuação obtida pela análise de sentimentos, podemos novamente utilizar nuvens de palavra como ferramenta para visualizar rapidamente as palavras normalmente associadas com sentimentos positivos ou negativos. A figura 5 mostra nuvens de palavras geradas utilizando especificamente conjuntos de texto que continham somente comentários positivos ou negativos, através de todos os *subreddits* analisados.

Figure 5. Nuvens de palavras geradas a partir de comentários positivos (esquerda) e negativos (direita)



As palavras predominantes nas nuvens de palavras positivas foram “hope” (esperança), “support” (suporte) e “slava ukraini”. A palavra “life” (vida) acabou se destacando na nuvem de palavras negativa, o que pode ser consequência do uso da palavra associado a sentimentos como lamentação ou luto.

4. Análise e detecção de comunidades

Esta seção tem como foco a apresentação dos grafos gerados a partir dos dados coletados de cada *subreddit* e posteriormente o processo de detecção de Comunidades dentro destes gráficos utilizando diferentes algoritmos.

4.1. Caracterização dos grafos

Os dados coletados de todos os *subreddits* tema deste trabalho foram usados para gerar um grafo referente a cada *subreddit*. Estes grafos são uma combinação dos dados de todo o período que a coleta abrangeu, de forma que cada uma dessas redes representam os usuários que participaram de cada *subreddit* dentro do período, juntamente com as interações (comentários) que ocorreram entre estes usuários.

A figura 6 mostra as estatísticas encontradas para grafos gerados a partir dos dados coletados de cada *subreddit*.

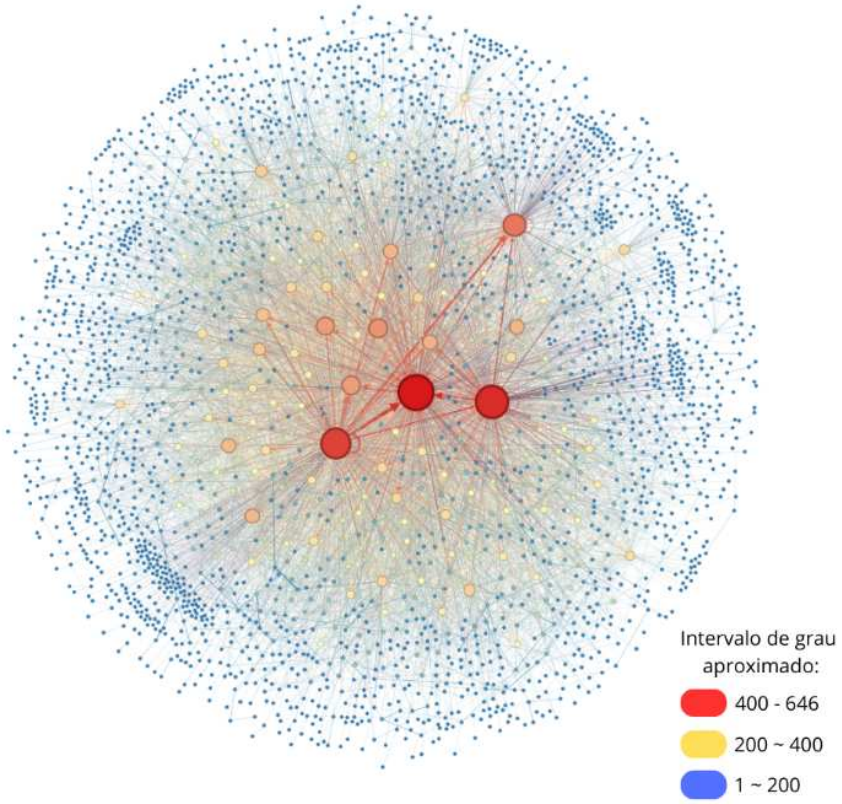
Figure 6. Estatísticas dos grafos

Subreddit	Nº de Nós	Nº de Arestas	Maior Grau	Grau médio	Diâmetro
r/EndlessWar	2581	12105	646	4,69	9
r/News	125862	275598	1692	2,19	27
r/Politics	99867	230426	2513	2,307	22
r/RussiaUkraineWar2022	31546	124361	3215	3,942	15
r/Ukraine	103289	431291	3680	4,176	23
r/UkraineWarVideoReport	85219	311319	6033	3,653	18
r/WorldNews	183845	437223	2720	2,37	20

A Figura 7 mostra a visualização do grafo gerado a partir dos dados coletados do *subreddit r/endlesswar*. Este grafo é dirigido, indicando a origem e o destino de cada

comentário por meio das setas nas pontas das arestas. Além disso, alguns elementos da visualização foram modificados para facilitar a compreensão. Primeiramente, o tamanho dos nodos cresce de acordo com seu grau total (soma da quantidade de arestas saindo e quantidade de arestas entrando no nodo). Similarmente, as arestas ficam maiores com base em seu peso (quantas interações ela representa). Por fim, o grafo foi colorido também baseando-se no grau dos nodos, onde pontos azulados representam um grau pequeno, enquanto pontos mais alaranjados ou avermelhados representam um grau relativamente alto comparado com o restante da rede.

Figure 7. Grafo de r/endlesswar

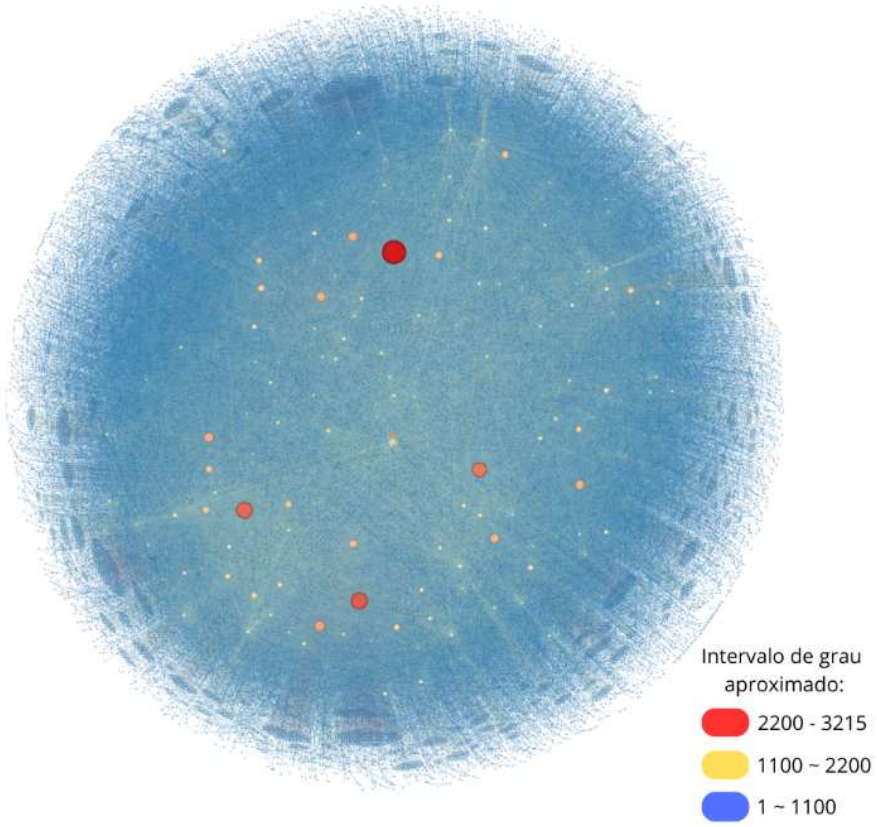


O grafo possui diversos pontos de alta densidade (*hubs*), representados pelos nodos mais avermelhados e maiores, que notavelmente possuem uma grande quantidade de arestas conectadas a eles. Isso indica usuários que enviaram muitas respostas para outros comentários, receberam muitas respostas em seus comentários, ou ambos. Como este *subreddit* em específico tem um tamanho pequeno comparado aos outros analisados neste trabalho, também é possível que esses usuários possuam alguma relação direta com o fórum, como por exemplo o cargo de moderador, e consequentemente interagem de forma frequente o suficiente para influenciar o tamanho de seus respectivos nodos no grafo.

A figura 8 mostra a visualização do grafo para *r/russiaukrainewar2022*. Um ponto curioso deste grafo, é que apesar do número muito menor de nodos comparado a alguns outros subreddits, o maior grau foi muito elevado. Isso indica que, em algum momento da Comunidade dentro do intervalo de tempo dos dados coletados, o usuário representado

pelo maior nodo do grafo, destacado em vermelho, obteve muita repercussão em suas postagens ou comentários.

Figure 8. Grafo de r/russiaukrainewar2022



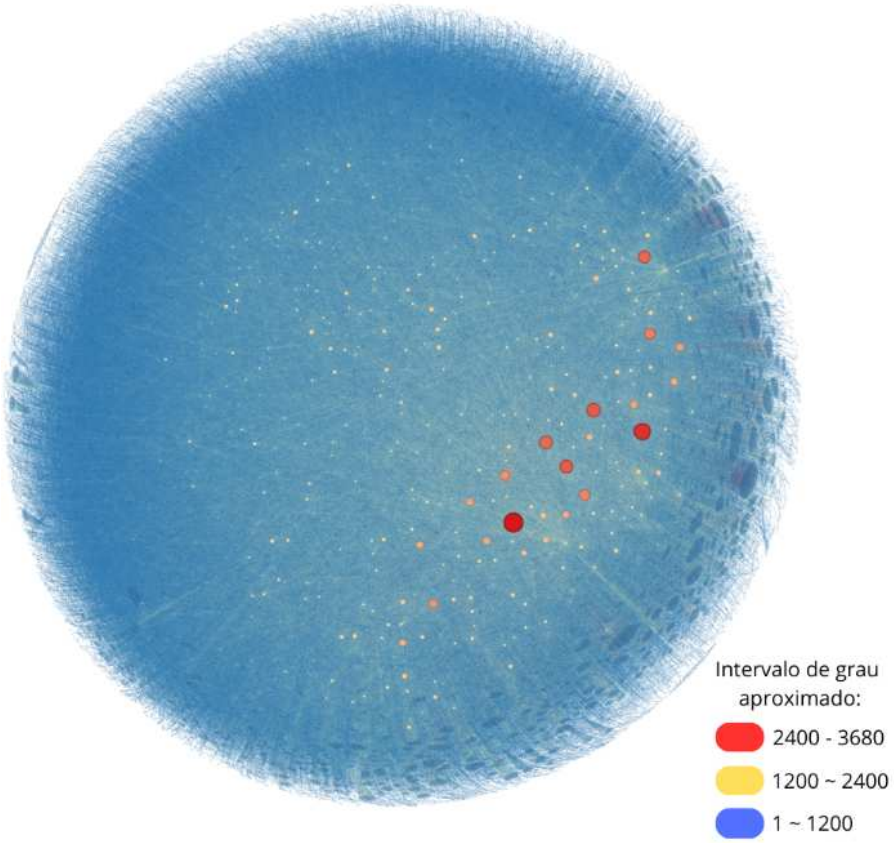
A distribuição dos nodos de grau mediano também denota uma região do grafo com grande quantidade de interações. É perceptível a concentração desses nodos na área central-inferior, possivelmente indicando que estes usuários interagiram muito frequentemente entre si.

A figura 9 mostra a visualização do grafo para *r/ukraine*. Esse *subreddit* tem uma quantidade de arestas altíssima comparado com os outros analisados neste trabalho. De fato, observando a visualização do grafo, fica claro a grande densidade de interações que ocorrem durante o período de coleta dos dados neste fórum. É importante explicar aqui o contexto desta Comunidade dentro da rede social para analisar este grafo. O *subreddit* "r/Ukraine", desde o início da guerra se tornou um grande centro de discussão e compartilhamento de informação sobre ela, não apenas no Reddit como também na internet como um todo. Essa página recebeu muitos papéis diferentes com o conflito, dentre eles: *feed* de notícias relacionadas a guerra, local de mobilizações sociais para apoio aos ucranianos, fórum de discussão política, militar e humanitária etc. Possivelmente, foram essas as razões que causaram uma concentração tão grande de interações neste grafo, comparado aos outros neste trabalho.

É perceptível que a maioria dos *hubs* ficaram na mesma região do grafo, mais para a metade direita. Baseando-se na ideia de que usuários frequentadores do fórum no

mesmo intervalo de tempo tem maior probabilidade de se conectarem, podemos inferir que essa concentração de nodos que ocorre no grafo pode ter acontecido devido ao imenso fluxo de usuários novos para a Comunidade (que anteriormente a guerra servia mais como um local direcionado para ucranianos) quando o conflito começou, que posteriormente deixaram de frequentar as discussões e por isso não criaram conexões através de toda a extensão do grafo.

Figure 9. Grafo de r/ukraine

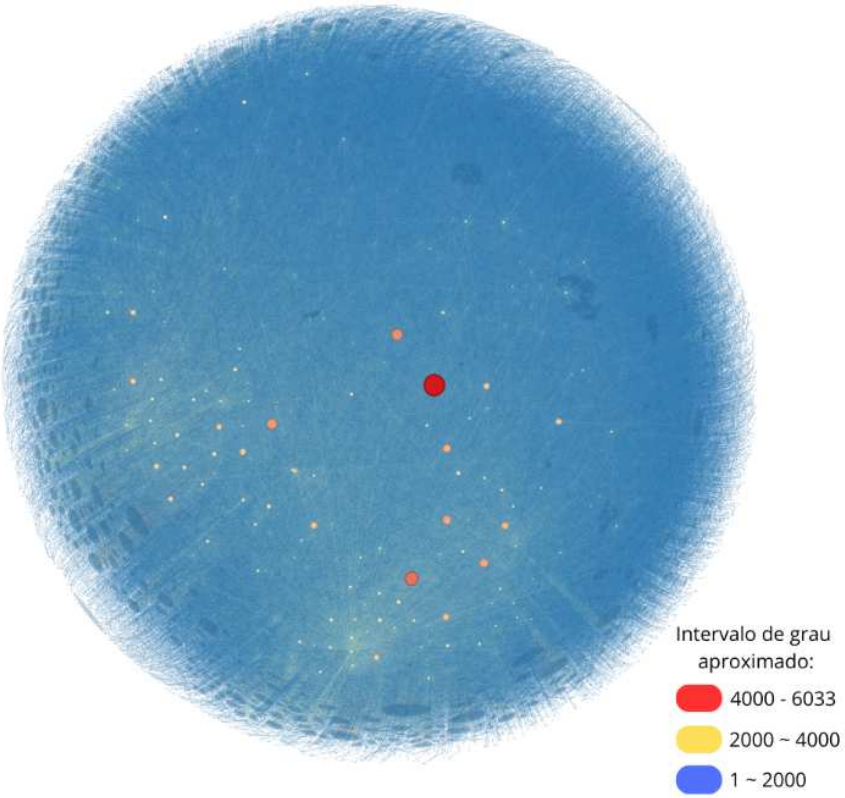


A figura 10 mostra a visualização do grafo para *r/ukrainewarvideoreport*. Vemos que o grafo no geral não tem muitos nodos no intervalo de grau mais elevado, apesar de os *hubs* terem atingido valores de grau bem altos. Uma possível explicação é pelo fato de as postagens do *subreddit* serem, em sua maior parte, compostas por vídeos e imagens. Nesse cenário, cada postagem pode receber uma grande quantidade de respostas (principalmente se o conteúdo exposto nelas for chocante) mas ao mesmo tempo não provocar muita discussão (réplicas e tréplicas) entre os usuários, tornando a rede menos densa.

4.2. Detecção de comunidades

Esta seção apresenta os resultados da execução de algoritmos de detecção de Comunidades sobre os grafos gerados previamente. O intuito desta parte do trabalho é observar o comportamento de cada algoritmo e compará-los entre si, além de mostrar as Comunidades encontradas de forma visual.

Figure 10. Grafo de r/ukrainewarvideoreport



4.2.1. Algoritmo de Louvain

A execução desse algoritmo busca encontrar a melhor "partição" de um grafo, medida pelo seu valor de modularidade. Assim, a partição contém as diversas Comunidades que foram encontradas. Para fins de facilitar a análise neste trabalho, foram selecionadas as 6 maiores Comunidades encontradas em cada grafo (baseado em seu número de nodos) e geradas as visualizações destas Comunidades.

A Figura 11 mostra as estatísticas das 6 maiores comunidades de "r/endlesswar". Algumas informações interessantes podem ser extraídas da Tabela, como o fato de algumas Comunidades menores, terem a média de grau não tão distante de Comunidades maiores, como a 5 e a 2. Possivelmente, isso ocorre quando usuários presentes nessa Comunidade tiveram discussões mais extensas entre si, gerando um grau elevado mesmo que a quantidade de participantes na discussão seja menor.

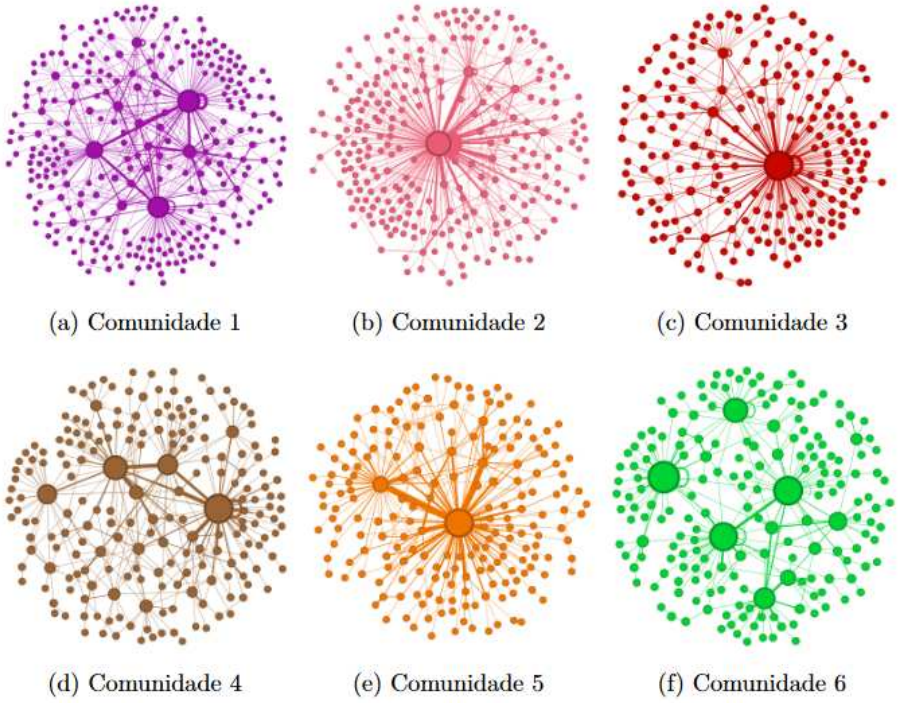
Além disso, vemos que a Comunidade 1 possui um grau máximo bem menor que a Comunidade 2, mas ainda obteve uma média muito maior, indicando uma concentração maior de usuários com grau alto. Considerando a maneira que o Reddit é estruturado, é possível que usuários que interagiram com outros usuários que possuem grau alto de interações também tenham uma quantidade de interações elevada. Isso ocorre pela fato de postagens e comentários que tem grande quantidade de votos positivos no fórum ficarem em destaque na página, o que implica que as réplicas também ficarão destacadas junto, e terão uma visibilidade maior.

Figure 11. Estatísticas das comunidades de r/endlesswar

Comunidade	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	292	476	3,25	78	7
2	262	390	2,97	165	6
3	205	262	2,55	123	9
4	203	294	2,89	51	8
5	197	278	2,82	103	8
6	185	232	2,5	38	9

A Figura 12 mostra os grafos gerados a partir de cada uma das Comunidades. Essa visualização proporciona uma compreensão melhor da estrutura de cada Comunidade. É notável que em alguns casos, como ocorre nas Comunidades 2, 3 e 5, o algoritmo parece ter detectado um agrupamento centrado em apenas um ou dois usuários *hubs*, possivelmente indicando discussões de grande repercussão provocadas por estes usuários.

Figure 12. Visualização das comunidades de r/endlesswar



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

A Figura 13 mostra as estatísticas para as 6 maiores Comunidades encontradas pelo algoritmo de Louvain em "r/russiaukrainewar2022". O grau médio teve uma diferença considerável entre a maior e a menor Comunidade. Além disso, o grau máximo não foi necessariamente proporcional ao tamanho da Comunidade, diferentemente de grafos anteriores. O diâmetro, por sua vez, foi relativamente parecido entre todas as Comunidades.

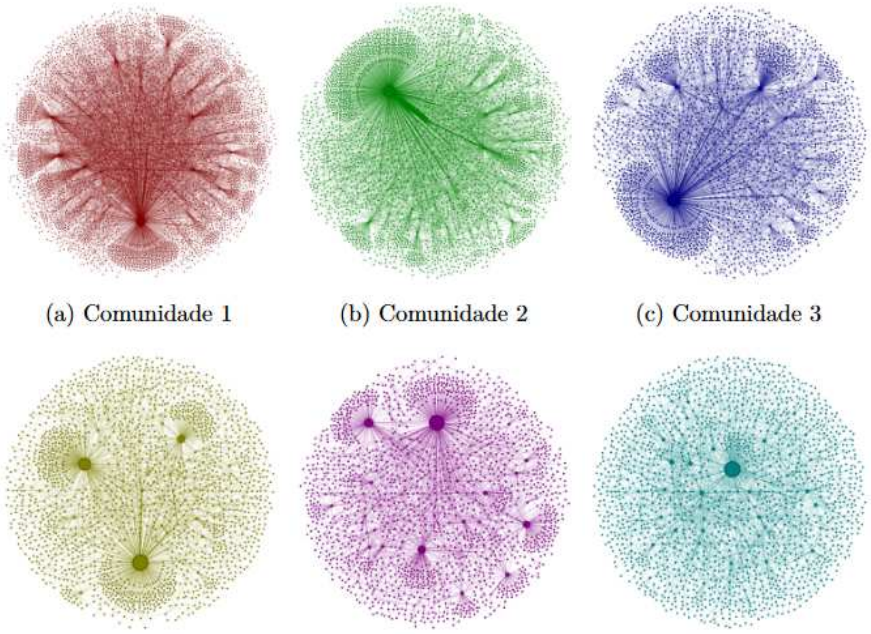
Em relação aos grafos de cada Comunidade, observados na Figura 14, vemos Comunidades com uma grande quantidade de *hubs*, como a 1 e 3, juntamente com Comunidades de poucos *hubs* como a 6, onde apenas alguns poucos nodos tem tamanho de destaque devido ao seu grau.

A Figura 15 mostra as estatísticas encontradas para as 6 maiores Comunidades

Figure 13. Estatísticas das comunidades de r/russiaukrainewar2022

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	4194	7683	3,66	971	9
2	3117	4934	3,16	1194	9
3	2507	3749	2,99	835	10
4	1850	2566	2,77	408	11
5	1809	2573	2,84	360	10
6	1731	2556	2,95	234	12

Figure 14. Visualização das comunidades de r/russiaukrainewar2022



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

detectadas pelo algoritmo de Louvain em "r/ukraine". Dessa vez, a quantidade de nodos é extremamente diferente da primeira até a última Comunidade, e a proporção do número de arestas não foi necessariamente proporcional ao número de nodos. O grau médio também não foi proporcional, como é possível notar observando a Comunidades 2, por exemplo. O grau máximo também foi extremamente variável, de forma que até mesmo Comunidades menores obtiveram um valor comparativamente alto. O diâmetro de todas as Comunidades se manteve bem parecido.

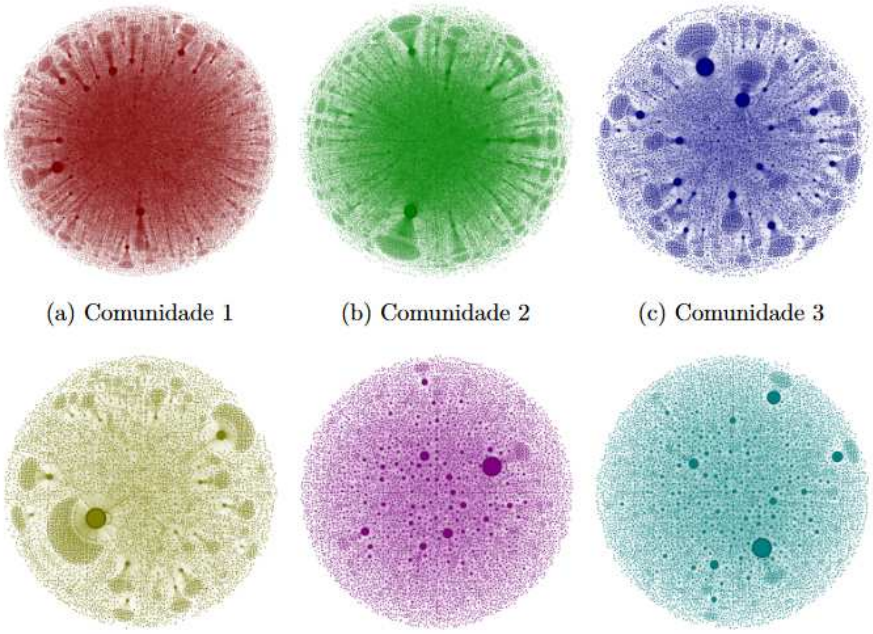
Figure 15. Estatísticas das comunidades de r/ukraine

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	16772	31092	3,7	692	13
2	11327	23280	4,11	1505	11
3	7512	11628	3,09	760	13
4	6651	9289	2,79	1181	15
5	6346	10567	3,33	296	13
6	6310	10399	3,29	283	13

Observando os grafos de cada Comunidade na Figura 16, é notável que no geral cada Comunidade teve vários *hubs*, ao invés de apenas um só. Isso significa que o algoritmo não detectou Comunidades que parecem ser centradas em apenas um usuário com

grande repercussão, mas sim em diversos usuários diferentes que participaram ativamente de determinada discussão e obtiveram um grande número de interações.

Figure 16. Visualização das comunidades de r/ukraine



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

.

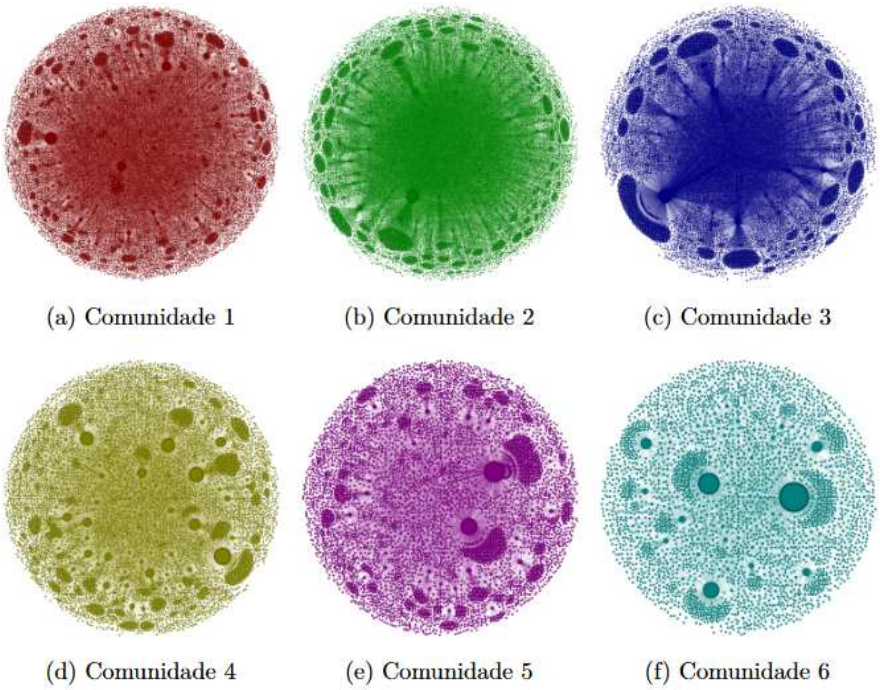
A Figura 17 mostra as estatísticas para as 6 maiores Comunidades encontradas pelo algoritmo de Louvain em "r/ukrainewarvideoreport". Existe grande variação entre o número de nodos e número de arestas. Além disso, o grau médio também difere bastante, e as Comunidades 2 e 3 tiveram grau médio maior do que a Comunidade 1, apesar de serem menores em quantidade de nodos. O grau máximo das Comunidades também variou bastante, com a Comunidade 3 tendo um valor muito acima do restante das Comunidades.

Figure 17. Estatísticas das comunidades de r/ukrainewarvideoreport

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	16645	29284	3,51	837	13
2	14650	31247	4,26	1972	10
3	11921	24039	4,03	2743	10
4	8888	14923	3,35	590	12
5	6463	9811	3,03	900	13
6	3502	4578	2,61	480	14

A visualização dos grafos desta Comunidade, presentes na Figura 18, mostram que de forma geral cada Comunidade teve diversos *hubs*, bem distribuídos através de toda a extensão de cada grafo. Isso indica que, dentro de cada Comunidade, diversos usuários diferentes obtiveram grande repercussão em suas postagens e comentários. As três maiores Comunidades, em particular, parecem ter uma estrutura extremamente semelhante: diversos *hubs* espalhados por todas as direções dos grafos, próximos às bordas.

Figure 18. Visualização das comunidades de r/ukrainewarvideoreport



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

4.2.2. Algoritmo de Leiden

Assim como o algoritmo de Louvain, o algoritmo de Leiden foi executado sobre as Comunidades *subreddits* analisadas neste trabalho. A Figura 19 fornece as estatísticas das 6 maiores Comunidades detectadas pelo algoritmo de Leiden no *subreddit* "r/endlesswar". A Comunidade 1 possui um número de nodos relativamente elevado comparado as outras, que possuem valores num intervalo menor. O número de arestas também é bastante elevado na primeira Comunidade, mas com variação menor a partir da Comunidade 3 até a 6. As Comunidades 1, 2 e 6 tiveram valores mais elevados de grau médio. Por fim, o diâmetro é similar entre todas as Comunidades.

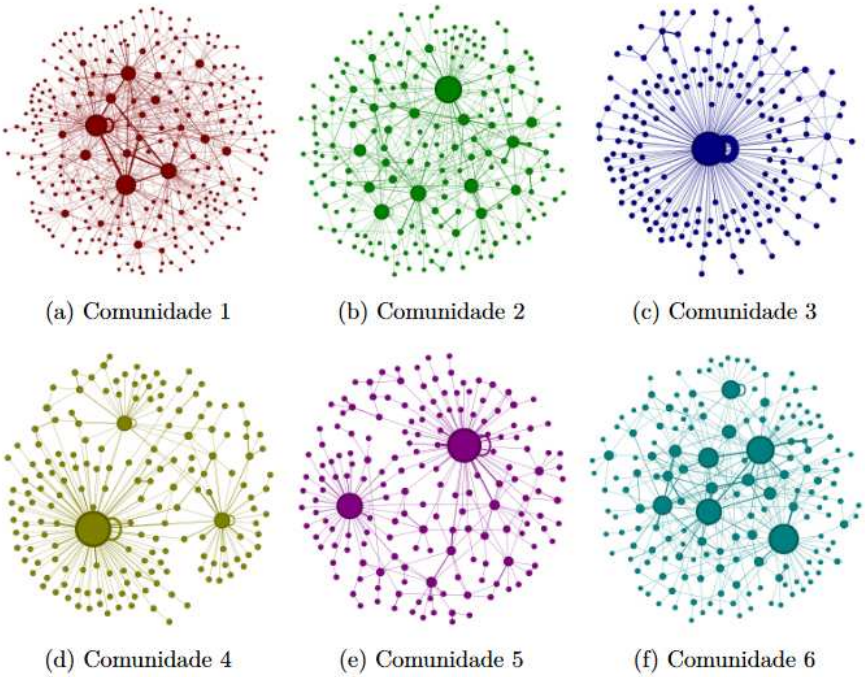
Figure 19. Estatísticas das comunidades de r/endlesswar

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	262	556	4,24	45	6
2	196	379	3,66	34	7
3	180	209	2,32	31	6
4	171	202	2,36	87	7
5	170	219	2,57	36	8
6	165	299	3,62	20	6

Na Figura 20 vemos a visualização dos grafos detectados por Leiden neste *subreddit*. De forma geral, os grafos não ficaram centradas em apenas um usuários, mas sim em vários pontos com grau mais elevado, como é perceptível nas Comunidades 1, 2 e 6, por exemplo.

A Figura 21 mostra o grafo das estatísticas encontradas para as 10 maiores Comunidades detectadas pelo algoritmo de Leiden em "r/RussiaUkraineWar2022". O número de nodos decresce rapidamente das duas primeiras Comunidades para as outras. Todas

Figure 20. Visualização das comunidades de r/endlesswar



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

.

as Comunidades a partir da Comunidade 4 tiveram um tamanho relativamente próximo, ainda que o número de arestas seja mais variável entre elas. O grau médio também não foi proporcional ao tamanho de cada Comunidade, mas o valor desse dado em todas as Comunidades desse *subreddit* foi relativamente elevado, comparando-o às estatísticas de outros *subreddits*. O diâmetro de todas as Comunidades foi extremamente parecido.

Figure 21. Estatísticas das comunidades de r/russiaukrainewar2022

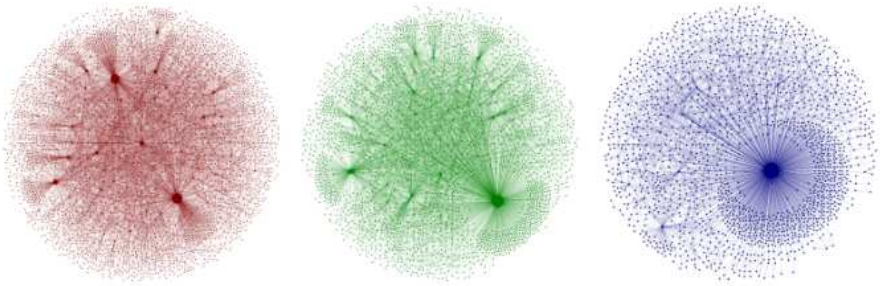
Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	3562	6887	3,86	515	10
2	2794	5028	3,59	928	9
3	1767	2553	2,89	1019	10
4	1205	1656	2,74	661	10
5	1140	1675	2,93	317	9
6	1050	1636	3,11	208	11

Observando a Figura 22, que mostra os grafos gerados a partir de cada Comunidade, é perceptível que algumas Comunidades tiveram uma dispersão do grau dentre a extensão da rede, com mais *hubs* aparecendo em cada grafo.

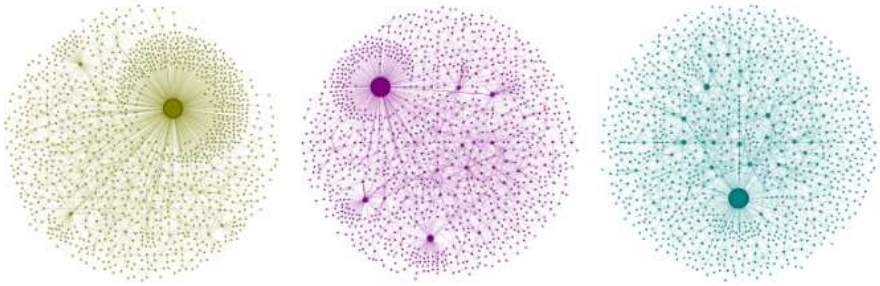
A Figura 23 mostra as estatísticas das Comunidades encontradas pelo algoritmo de Leiden no *subreddit* "r/ukraine". As 3 primeiras Comunidades obtiveram um grande número de nodos comparado a Comunidades de *subreddits* anteriores. O número de arestas em geral foi proporcional ao tamanho de cada Comunidade, com exceção das Comunidades 2 e 3. As 4 primeiras Comunidades também tiveram um grau médio bem elevado, um sinal de que as discussões que aconteceram nesse *subreddit* provavelmente possuíram uma quantidade acima da média de réplicas e tréplicas nos comentários.

A Figura 24 revela os grafos gerados a partir das Comunidades que foram detec-

Figure 22. Visualização das comunidades de r/russiaukrainewar2022



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3



(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

.

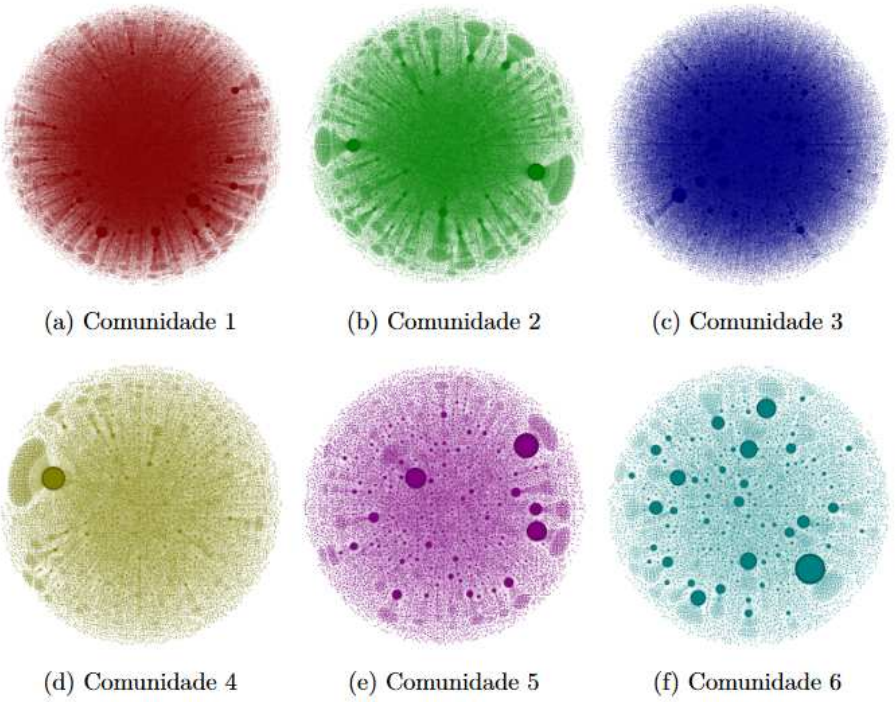
Figure 23. Estatísticas das comunidades de r/ukraine

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	18895	44449	4,7	837	11
2	13942	27213	3,9	1674	10
3	13217	36670	5,54	342	11
4	7796	14354	3,68	1365	11
5	7116	12437	3,49	494	13
6	5519	8211	2,97	347	12

tadas pelo algoritmo de Leiden nesse *subreddit*. As 3 primeiras Comunidades ficaram visualmente bem densas, uma consequência da quantidade grande de número de nodos somado ao valor alto de grau médio, fazendo com que os nodos no geral tenham um tamanho maior no grafo.

A Figura 25 mostra as estatísticas das Comunidades detectadas pelo algoritmo de Leiden para o *subreddit* "r/ukrainewarvideoreport". Assim como o *subreddit* anterior, este também obteve valores bem altos de número de nodos, número de arestas e grau médio particularmente nas Comunidades 1, 2 e 3.

Figure 24. Visualização das comunidades de r/ukraine



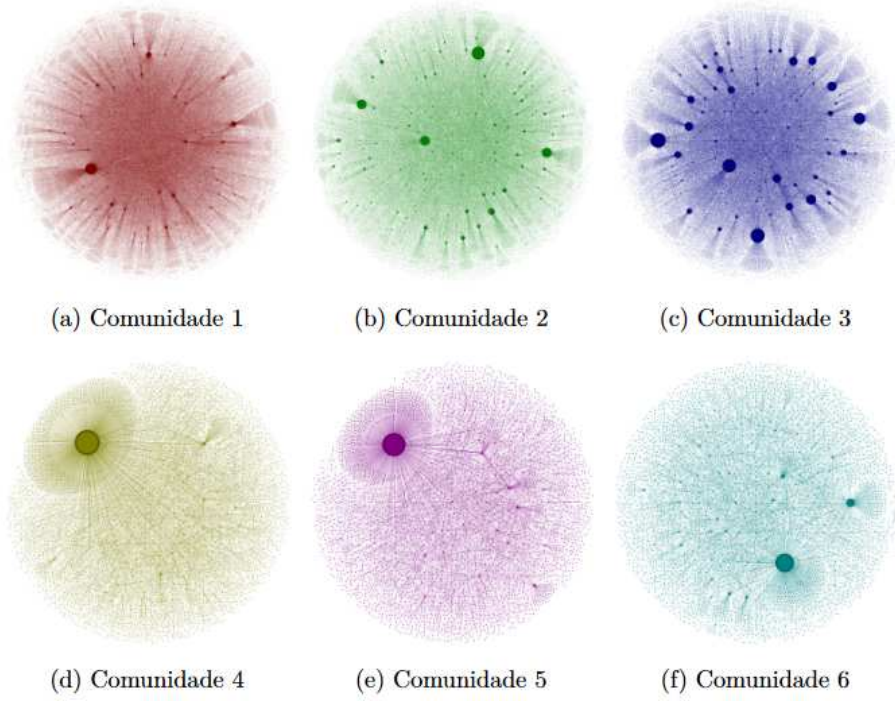
(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

Figure 25. Estatísticas das comunidades de r/ukrainewarvideoreport

Comunidades	Nº Nodos	Nº Arestas	Grau médio	Grau máximo	Diâmetro
1	15011	37738	5,02	2216	9
2	14129	28450	4	898	9
3	10085	20422	4,05	726	10
4	4847	7123	2,93	1970	11
5	3928	5516	2,8	1098	13
6	3790	5435	2,86	600	14

Figure 26. Visualização das comunidades de r/ukrainewarvideoreport



(a) Comunidade 1 (b) Comunidade 2 (c) Comunidade 3

(d) Comunidade 4 (e) Comunidade 5 (f) Comunidade 6

A Figura 26 revela os grafos que foram gerados a partir das Comunidades detectadas para este *subreddit*. As 3 primeiras Comunidades tiveram uma grande quantidade de *hubs* diferentes, o que é esperado dado seus valores elevados de grau médio. Por outro lado, as Comunidades 4 e 5 tiveram apenas um usuário de destaque, enquanto o resto do grafo possui nodos com grau mais baixo, o que explica os seus valores comparativamente menores de grau médio e ao mesmo tempo um alto grau máximo.

5. Considerações finais e trabalhos futuros

A etapa inicial deste trabalho consistiu na extração e posterior caracterização de dados, para auxiliar o objetivo futuro de aplicar algoritmos de detecção de comunidades. Para este fim, foram aplicadas algumas técnicas de análise, como a análise de sentimentos e a análise psicolinguística , bem como a apresentação visual do conjunto de dados utilizando gráficos e nuvens de palavra.

As principais dificuldades nesta etapa do trabalho foram o estudo e aplicação das ferramentas de análise de sentimentos e psicolinguística, e também a elaboração de alguns gráficos apresentados, como as nuvens de palavras e a rede de co-ocorrência. Essas dificuldades surgiram, em parte, da necessidade de adaptar os dados extraídos do Reddit para que funcionassem com os algoritmos para análise e para gerar os gráficos.

Posteriormente, os dados coletados também foram utilizados para a geração de grafos. Os grafos gerados consistem de nodos representando os usuários de cada *subreddit*, e arestas representando interações entre esses usuários. A partir destes grafos foi possível extrair diversas estatísticas de cada *subreddit*, como o número de usuários participantes e quantas vezes cada usuários interagiu dentro da comunidade. Os resultados obtidos nessa etapa foram úteis para realizar análises dos comportamentos e fenômenos sociais ocorridos nestes fóruns de discussão.

Utilizando os algoritmos de detecção de comunidades de Louvain e de Leiden, foram gerados grafos das 10 maiores comunidades detectadas por cada algoritmo para os *subreddits* mais relevantes ao conflito entre a Rússia e a Ucrânia. Essas comunidades, por sua vez, também forneceram estatísticas importantes para compreender melhor as interações entre usuários e a interconectividade entre os participantes de cada comunidade. Além disso, foi possível comparar os resultados dos dois algoritmos executados e entender melhor suas semelhanças e diferenças.

O desafio principal desse ponto do trabalho foi compreender as implicações de cada estatística obtida a partir dos grafos gerados e da estrutura de cada grafo. Fatores como a quantidade de interações média dos usuários dentro de uma comunidade ou a posição de um nodo em um grafo possuem significados diferentes dependendo do contexto do *subreddit* de origem. Assim, além do estudo da teoria de grafos e do funcionamento dos algoritmos de detecção, também foi necessária uma compreensão do Reddit como ferramenta social, com uma cultura própria que influenciou todos os resultados obtidos.

A partir da comparação dos resultados ao final do desenvolvimento, ficou evidente que, de forma geral, as comunidades detectadas por cada algoritmo foram semelhantes em relação aos temas que mais foram discutidos e mais geraram repercussão, ainda que os usuários de maior grau em cada comunidade diferiram por algoritmo utilizado. Os *subreddits* que foram analisados neste trabalho se mostraram, em grande parte, apoiadores do

lado ucraniano do conflito, com uma notável exceção sendo o *subreddit* "r/EndlessWar", que apresentou um ponto de vista mais neutro, criticando ambos os países em guerra.

Em relação a trabalhos futuros, cabem as seguintes propostas:

- Executar outros algoritmos de detecção de comunidades no conjunto de dados coletados;
- Identificar outros *subreddits* relevantes ao tópico deste trabalho que possam aumentar a diversidade do material analisado;
- Delimitar intervalos de tempo menores nos dados coletados, para realizar análises comparativas de períodos diferentes;
- Realizar análise de sentimentos e psico-linguística para as comunidades detectadas por cada algoritmo.

References

[Anderson 2015] Anderson, K. (2015). Ask me anything: what is reddit? *Library Hi Tech News*, 32:8–11.

[Bedi and Sharma 2016] Bedi, P. and Sharma, C. (2016). Community detection in social networks. *WIREs Data Mining and Knowledge Discovery*, 6(3):115–135.

[Chunaev 2020] Chunaev, P. (2020). Community detection in node-attributed social networks: A survey. *Computer Science Review*, 37:100286.

[Ciuriak 2022] Ciuriak, D. (2022). The role of social media in russia's war on ukraine.

[Elena-Iulia 2018] Elena-Iulia, V. (2018). The Importance Of Social Media. *Annals - Economy Series*, 6:80–91.

[Fortunato 2010] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

[Hintz and Betts 2022] Hintz, E. A. and Betts, T. (2022). Reddit in communication research: current status, future directions and best practices. *Annals of the International Communication Association*, 46(2):116–133.

[Newman and Girvan 2004] Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113.

[Proferes et al. 2021] Proferes, N., Jones, N., Gilbert, S., Fiesler, C., and Zimmer, M. (2021). Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Social Media + Society*, 7(2):20563051211019004.

[Zhu et al. 2022] Zhu, Y., ul Haq, E., Lee, L.-H., Tyson, G., and Hui, P. (2022). A reddit dataset for the russo-ukrainian conflict in 2022.