METODOLOGIA DE IDENTIFICAÇÃO DE POLARIDADE EM TEXTOS COM BASE EM PROJETOS DE LEI BRASILEIROS

Daniel Lopes Braz dos Santos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Civil.

Orientador: Nelson Francisco Favilla Ebecken

Rio de Janeiro

Dezembro de 2017

METODOLOGIA DE IDENTIFICAÇÃO DE POLARIDADE EM TEXTOS COM BASE EM PROJETOS DE LEI BRASILEIROS

Daniel Lopes Braz dos Santos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

_____

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

_____

Prof. Claudia Medina Coeli, D.Sc.

_____

Pesq. Myrian Christina de Aragão Costa, D.Sc.

RIO DE JANEIRO, RJ BRASIL

DEZEMBRO DE 2017

Santos, Daniel Lopes Braz dos.

Metodologia de Identificação de Polaridade em Textos com Base em Projetos de Lei Brasileiros/ Daniel Lopes Braz dos Santos. – Rio de Janeiro: UFRJ/COPPE, 2017.

XI, 47 p.: il.; 29,7 cm.

Orientador: Nelson Francisco Favilla Ebecken.

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Civil, 2017.

Referências Bibliográficas: p. 46-47.

1. Aprendizado de Máquina. 2. Mineração de Textos. 3. Clusterização. 4. Polarização de Textos. I. Ebecken, Nelson Francisco Favilla. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Agradecimentos

Gostaria de agradecer primeiramente a Deus por toda a força em todos os momentos que precisei. Agradecer a minha família e meus amigos por todo amparo nos momentos de dificuldade e na companhia nos momentos de alegria. Em especial a Mirela que aguentou todo esse final de mestrado comigo, os estresses, a falta de tempo e sempre me ajudou quando precisei. Agradeço a meu Orientador Nelson por todo o apoio, ajuda e conselhos durante toda essa Jornada. À Myrian e Valeria por toda a ajuda, aos dias de correria e de total dedicação e apoio para me ajudar quando precisei, além de todos os momentos de alegria e descontração. Agradeço por todo o carinho e apreço que sempre demonstraram comigo.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

METODOLOGIA DE IDENTIFICAÇÃO DE POLARIDADE EM TEXTOS COM BASE EM PROJETOS DE LEI BRASILEIROS

Daniel Lopes Braz dos Santos

Dezembro/2017

Orientador: Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

Com o aumento da discussão e interesse sobre política, tanto pela sociedade como por especialistas, surge a necessidade de se analisar, estudar e acompanhar as ações realizadas pelo legislativo e executivo. Mas o grande volume de documentos, projetos de lei, assim como seu tamanho e padrão de escrita e *layout*, acabam dificultando este processo. Vê-se a necessidade de uma ferramenta que consiga identificar e agrupar cada tema de projeto e separar estes documentos quanto as suas polaridades em relação ao tema identificado.

O objetivo desta dissertação é apresentar a pesquisa de uma metodologia eficiente de identificação de polaridade e similaridade de documentos sobre um mesmo tema, usando como estudo de caso projetos de lei contra e a favor da liberalização do aborto no Brasil. Busca-se primeiramente identificar e comparar técnicas de aprendizado de máquina sobre textos que consigam classificar os projetos nestes dois vieses citados. Esta metodologia analisa cada parte do projeto em questão, utilizando a técnica de agrupamento K-means, aplicando em seguida um método baseado em grafos para processar todas as combinações de parâmetros e verificar os projetos que possuem maiores ligações entre si. Para a realização de testes foram utilizados projetos previamente classificados e os resultados obtidos com esta metodologia demonstraram ligações e peculiaridades muito interessantes e promissoras, ajudando assim, na identificação de similaridades e padrões em documentos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

TEXT POLARITY IDENTIFICATION METHODOLOGY BASED ON BRAZILIAN DRAFT LAWS

Daniel Lopes Braz dos Santos

December/2017

Advisor: Nelson Francisco Favilla Ebecken

Department: Civil Engeneering

With the increase of discussion and interest in politics, both by society and by specialists, comes up the need to analyze, study and follow the legislative's and executive's actions. But the large number of documents, draft laws, as well as their layout and writing pattern, makes this a difficult process. The necessity of identifying and clustering each draft law subject and separate these documents by their polarities in relation to the identified theme.

The goal of this dissertation is to build an efficient methodology for document polarity and similarity identification, over the same subject. Brazilian draft laws on abortion liberalization were analyzed as case study. The first step is to identify and compare different machine learning techniques the can sort law projects on both against and in favor bias. The methodology analyses each part of the document, utilizing K-means grouping, to later apply a graph based process on every parameter combination certify the drafts have more connection among each other. To perform tests, previously classified draft laws manually sorted by a specialist were used and all results obtained with this method demonstrated very interesting and promising connections and peculiarities, thus helping to identify similarities and patterns in documents.

Sumário

1. Introdução 1

1.1 Motivação 2
1.2 Revisão 2

2. Metodologia 3

2.1 Pré-processamento 3

2.1.1 Particionamento do documento 4
2.1.2 Remoção de *Stopwords* 5
2.1.3 Representação em *Bag-of-Words* 5
2.1.4 Ponderação de Termos (TF-IDF) 8
2.1.5 N-Gram 10
2.1.6 Scikit-learn 12

2.2 Agrupamento (Clusterização) 12

2.2.1 Combinação de parâmetros 13

2.3 Visualização em Grafos 17

2.3.1 Coloração de vértices 17
2.3.2 Ponderação de arestas 18
2.3.3 Top Grafos 20

3. Estudo de caso 21

3.1 Pré-processamento 21
3.2 Visualização de resultados 25

3.2.1 Visualização em grafos 26

4. Análise de Resultados 35
5. Conclusões 43

5.1 Trabalhos Futuros 44

Referências 46

Lista de Figuras

Figura 1: Etapas da metodologia 3
Figura 2: Etapas do pré-processamento 4
Figura 3: Exemplo de processamento por bigram 11
Figura 4: Combinação de parâmetros 14
Figura 5: Clusterizações geradas pela combinação E2 16
Figura 6: Grafo gerado da combinação anterior 17
Figura 7: Coloração de vértices de acordo com o agrupamento (a) e (b) e de acordo com a polaridade (c) 18
Figura 8: Grafo com arestas ponderadas 19
Figura 9: Grafo com corte de 100% 20
Figura 10: Exemplo de projeto de lei 22
Figura 11: Projeto apensado 23
Figura 12: Combinações realizadas no estudo de caso 25
Figura 13: Análise de cenário da combinação C1 26
Figura 14: Grafo da combinação C2 com coloração em relação ao cluster K=5 27
Figura 15: Grafo da combinação C2 com coloração em relação ao cluster K=6 27
Figura 16: Representação em grafo da combinação C2 28
Figura 17: Destaque do grupo independente da Figura 16 29
Figura 18: Exemplo de combinação de seis execuções com arestas ponderadas 30
Figura 19: Normalização das arestas da combinação de seis execuções 31
Figura 20: Arestas ponderadas da Figura 16 31
Figura 21: Representação em grafo da combinação C3 32
Figura 22: Grafo C3 com corte 100% 33
Figura 23: Representação em grafo da combinação C3 com corte de 80% 33
Figura 24: Representação em grafo da combinação C3 com corte de 70% 34
Figura 25: Representação em grafo da combinação C2 com destaque para alguns PLs 35
Figura 26: Destaque do grupo independente da Figura 25 36
Figura 27: Grafo C3 com corte 100% com destaque para os dois pares 37
Figura 28: Combinação C3 com corte 70% - destaque para PLs 38
Figura 29: Combinação C3 com corte 80% - destaque para PLs 7 e 8 40
Figura 30: Destaque do grupo independente da Figura 28 41

Figura 31: Diagrama de áreas de assunto de PLs 42
Figura 32: Combinação C1 com destaque para o grupo 3..... 44

Lista de Tabelas

Tabela 1: Representação de uma Bag-of-words (BoW) 6
Tabela 2: Comparação dos dois corpora 7
Tabela 3: Combinação E1{K=(2:3); Textual(0:1);N-gram=(1:3)} 15
Tabela 4: Contagem de positivos e negativos da combinação E2 16
Tabela 5: Amostra de projetos de lei rotulados 21
Tabela 6: Quantidade e polaridade rotulada dos projetos de lei 21
Tabela 7: Valores da variável “Textual” 24

Lista de Abreviaturas

BoW	*Bag-of-Words*, p. 6
IDF	*Inverse Document Frequency*, p. 8
NLTK	*Natural Language Toolkit*, p. 5
PL	Projeto de Lei, p. 21
SUS	Sistema Único de Saúde, p. 38
SVM	*Support Vector Machine*, p. 12
TF	*Term Frequency*, p. 8

Capítulo 1

1. Introdução

A Política é um dos tópicos mais discutidos e estudados no Brasil e no mundo. Consequentemente, surge a necessidade de se obter informações sobre projetos e ações do executivo e do legislativo, bem como os responsáveis por estes. Esta necessidade se apresenta tanto para a informação geral da população quanto para estudos acadêmicos. Mas, devido ao grande número de documentos disponíveis, tais como os projetos de lei, sendo estes às vezes extensos, torna-se fundamental a utilização de ferramentas que automatizem as análises e sintetizem essas informações, viabilizando a democratização do seu acesso. Desta maneira será possível separar os projetos por temas, e dentro de cada tema identificar suas polaridades, ou seja, se são contra ou a favor de um determinado assunto, aumento ou diminuição de gravidades de crimes, entre outros. O foco deste trabalho é identificar polaridade em projetos de lei que possuem como tema o aborto. Define-se como polaridade a diferenciação dos projetos em relação ao seu posicionamento sobre o tema. Assim, busca-se separar os documentos em dois grupos, favoráveis e contrários à liberalização do aborto. Além disso, visa-se identificar as principais palavras e termos utilizados em cada grupo, bem como o quão interligados os projetos estão quanto ao seu tema e conteúdo.

Para este estudo foi utilizado o método de agrupamento K-means (Miner et al., 2012), implementado na ferramenta scikit-learn (2017). Este método foi escolhido para que fosse possível analisar o comportamento dos projetos e suas ligações. Posteriormente, estes resultados serão comparados com outros modelos supervisionados e não supervisionados.

Esta dissertação está dividida em cinco capítulos. No capítulo dois é descrita a metodologia utilizada. O capítulo três apresenta o estudo de caso, descrevendo a base de textos utilizada nesta análise e suas características. A análise dos resultados é apresentada no capítulo quatro e a conclusão e trabalhos futuros são discutidos no capítulo cinco.

1.1 Motivação

Este estudo visa desenvolver uma metodologia eficiente que possa ser utilizada para identificação de polarização de projetos de lei relacionados a qualquer tema, e não somente ao do estudo de caso levantado. Um exemplo dessa utilização seria a de especialistas em ciência política, que não precisarão mais classificar manualmente projetos de lei quanto a sua polaridade e nem identificar as palavras encontradas que reforcem cada classificação. Através dos resultados encontrados neste estudo, os projetos contrários ao aborto podem servir de base para "tomada de decisão" que sejam favoráveis à população, implementando políticas de melhoria da saúde e diminuição de violência à mulher.

1.2 Revisão

Diversos estudos sobre estado da arte relacionados ao propósito deste artigo foram analisados, como o de Liu et al. (2015), que identifica palavras e termos relevantes para posteriormente serem rotulados e agrupados, e de Sista & Srinivasan (2004) que propõe um método de polarização léxica para classificação de textos. Porém, devido à especificidade do objeto desta dissertação, que analisa projetos de lei redigidos em português, foi definido não se utilizar dos métodos anteriores e se testar e analisar uma nova abordagem, utilizando-se de ferramentas já reconhecidas de classificação de textos e identificação de termos relevantes e similares.

Capítulo 2

2. Metodologia

As sessões a seguir descrevem, como ilustrado na Figura 1, as etapas da metodologia desenvolvida nesta dissertação: O pré-processamento dos projetos de lei, seu agrupamento através do algoritmo K-means e formas de representação para a visualização e melhor análise destes resultados. Todas as etapas da Metodologia foram executadas utilizando como base a linguagem Python (Python, 2017) e suas bibliotecas.

Para desenvolver e testar a metodologia, é necessário um conjunto de documentos com um volume suficientemente grande, e todos eles, com seus rótulos pré-definidos, que também é chamado nesta dissertação como polaridade, pois, como descrito posteriormente no estudo de caso (capítulo três), tratam-se de dois rótulos com finalidades (intenções), em sua maioria, opostas. Rótulos são as classificações de cada documento de acordo com o objetivo do estudo. Por exemplo, se o objetivo é identificar documentos que falam "bem" ou "mal" sobre um determinado assunto (sendo este assunto o foco do conjunto de documentos), é preciso que estes documentos estejam individualmente pré-classificados em "bem" ou "mal", para que se confira a eficiência da metodologia e que, assim, sejam feitas as análises do processo.

Figura 1: Etapas da metodologia

2.1 Pré-processamento

Afim de se entender a estrutura dos documentos, uma amostra destes é analisada manualmente. Após identificar os padrões do tipo de documento estudado, expressões regulares são utilizadas com o objetivo de se extrair as partes relevantes para o estudo.

Quando se busca tratar de partes específicas de textos não estruturados, estas expressões são utilizadas para identificar padrões e fazer a quebra em blocos de um certo trecho. Por exemplo, caso se busque extrair todo o texto entre as expressões “O Congresso Nacional decreta:” e “Justificativa”, que no caso é, de forma simplificada para esta explicação, uma das regras utilizadas nesta dissertação, uma expressão regular englobando-as é utilizada. Este processamento também é utilizado para tratar casos de documento que não seguem o padrão identificado na maioria. Esta falta de padronização é uma das grandes dificuldades de se trabalhar com textos não estruturados. Documentos com seções ou estruturas diferentes, subtítulos de início de seções divergentes da maioria são alguns exemplos deste problema e que podem ser contornados através de tratamentos de linguagem natural, como, por exemplo, as expressões regulares já citadas.

A seguir são descritas cada etapa do pré-processamento, como exemplificado na Figura 2.

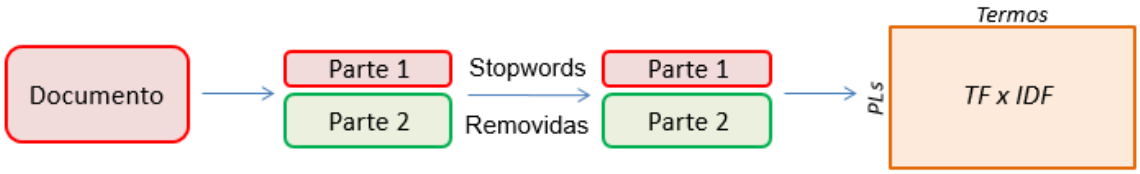

Figura 2: Etapas do pré-processamento

2.1.1 Particionamento do documento

Nesta etapa os documentos são lidos de forma ordenada pelo nome e salvos em memória com suas palavras reduzidas à forma caixa baixa (letras minúsculas). Em seguida, eles são quebrados em N partes, de acordo com a estrutura do texto e o objetivo do estudo. Por exemplo, caso se trabalhe com artigos científicos e se busque processar somente seus resumos e conclusões, estes serão identificados através das expressões regulares e os dois itens serão salvos para cada artigo. Uma variável chamada “Textual” deve ser criada para identificar qual parte dos documentos será processada. Utilizando ainda o exemplo anterior, a variável “Textual” teria três valores: 0, para processar somente o resumo, 1 somente a conclusão e 2 para processar o resumo e a conclusão juntos. Essa forma de analisar cada parte e a combinação dessas foi realizada com o intuito de se analisar o impacto de cada uma no agrupamento e no processo como um

todo, assim como analisar se alguma parte em especial identifica e representa melhor o seu respectivo texto, trazendo mais valor para ele no processo.

2.1.2 Remoção de *Stopwords*

No processamento de linguagem natural utiliza-se geralmente um grande volume de dados, textos e seus termos, logo a remoção de ruídos, dados irrelevantes ou que atrapalhem o processamento se tornam muito importante. As *stopwords* (Miner et al., 2012), são um exemplo disto, consistindo de uma lista de palavras ou termos que são removidas antes do processamento do texto por apresentarem baixo ou nenhum valor semântico. Esta lista é formada por termos muito frequentes em uma língua e palavras funcionais. Palavras funcionais são palavras utilizadas apenas para estabelecer relações entre elementos de um texto, como, por exemplo, artigos, conjunções e preposições. Advérbios também podem ser utilizados dependendo do propósito do trabalho. Porém não existe uma lista definida de stopwords. Algumas vezes, palavras muito frequentes em um grupo de textos a serem analisados também são incluídas nesta lista, pois não seriam muito úteis no processamento e poderiam inclusive atrapalhar. A remoção das *stopwords* depende do tipo de processamento que será realizado. Em casos onde a estrutura da frase é importante, a remoção delas poderia ocasionar na quebra desta estrutura, sendo maléfico para o processamento. Outro exemplo seria no caso de artigos definidos, que como o nome já diz, ajudam na identificação das entidades nomeadas, nomes próprios.

Para este estudo foi utilizada a lista de *stopwords* disponível no corpus "*Stopwords*" da biblioteca do Python NLTK (Bird et al., 2009). Esta biblioteca possui listas de palavras para diversos idiomas, inclusive para o português, com 203 palavras. Alguns exemplos destas palavras são: verbo "ter" e "ser" e suas flexões, artigos como "a" e "o" e preposições "de" e "para".

2.1.3 Representação em *Bag-of-Words*

No processamento de linguagem natural e recuperação da informação, formas de representação numérica do texto puro são comumente aplicadas. Certos algoritmos de classificação não conseguem trabalhar com processamento de texto no seu formato puro, precisando de uma transformação deste em dados numéricos como a representação em

vetor. O modelo *Bag-of-Words* (Harris, 1954), ou BoW, (quando os termos são unitários, ou seja, palavras) ou *Bag-of-Terms* é uma representação de textos através de vetores, em que é desconsiderada a ordem dos termos, e até mesmo sua estrutura gramatical na maioria das vezes. Ele consiste em uma matriz na qual suas linhas correspondem a cada documento e suas colunas a cada palavra (única), ou termo, do conjunto de palavras dos documentos processados ou de um conjunto de palavras pré-definidas através de um dicionário. Ou seja, cada documento $d_i \in D$, onde D representa o conjunto de documentos utilizados, é *tokenizado*, isto é, quebrado em palavras, ou termos, e se transforma em um vetor, $d_i = \{a_{i1}, a_{i2}, \dots, a_{ij}, \dots, a_{iN}\}$, onde N corresponde ao número total de termos únicos e a_{ij} corresponde ao peso do termo único j no documento i. Este valor de peso do termo é explicado melhor na próxima seção. O conjunto de palavras correspondente às colunas é também chamado de vocabulário. Para ilustrar melhor como funciona essa transformação, demonstra-se uma *bag-of-words* de três documentos, representada na Tabela 1, que neste caso correspondem às três frases simples a seguir:

Ex. 1: João gosta de futebol. Maria também gosta.

Ex. 2: Maria também gosta de assistir jogos de vôlei

Ex. 3: João não gosta de vôlei.

Tabela 1: Representação de uma Bag-of-words (BoW)

Ex:	**joão**	**gosta**	**de**	**futebol**	**maria**	**também**	**assistir**	**jogos**	**vôlei**	**não**
1	1	2	1	1	1	1	0	0	0	0
2	0	1	2	0	1	1	1	1	1	0
3	1	1	1	0	0	0	0	0	1	1

O número de palavras do vocabulário, em sua maioria, é bem extenso, transformando a matriz num conjunto gigante de dados. Porém, cada documento usa um número bem pequeno deste vocabulário, fazendo com que cada linha tenha em sua maioria elementos zerados. No exemplo acima, por se tratarem de apenas 3 exemplos, e com textos muito pequenos, o número de zeros ainda não é muito significativo, mas já é possível ter uma ideia de como, com um número maior de documentos e palavras dentro deles, essa matriz ficará com poucos dados preenchidos em comparação com o número de zeros. Utilizando como exemplo disto um corpora (conjunto de textos) muito utilizado para treinamento de classificadores de texto, o Movie Review (Pang et al., 2002), na qual

os documentos são textos de avaliações de filmes e está disponível na biblioteca NLTK, e uma base utilizada no estudo de Santos & Silva (2014), que lida com comentários de posts do Facebook (2017), pode-se visualizar melhor essas características da *bag-of-words*. Os dados de ambas estão representados na Tabela 2. Os dois corporas são bem diferentes no quesito de palavras por documento, enquanto o primeiro apresenta um número considerado de palavras, o segundo é bem mais curto. Esta disparidade de tamanho é útil para se verificar que apesar disto, o número de elementos nulos continua muito grande em ambos os casos. Pode-se observar que mesmo em textos maiores, o número de elementos nulos é gigante, correspondendo a 99,12% em média, considerando a média de palavras únicas por documento. E a medida que o número de documentos do corpora aumenta, o tamanho do vocabulário tende a crescer também.

Esta característica se chama matriz esparsa, ou seja, uma matriz com pouco valores não nulos. Este tipo de matriz é um problema pois ocupa muito espaço em memória, apesar de possuir poucos elementos que realmente possuem valor. Uma solução para isto é salvar somente os elementos não nulos, mas mantendo o acesso rápido aos dados. A biblioteca Scipy (Scipy, 2017) possui uma classe, a "sparse", usada nesta dissertação, que faz essa transformação, salvando somente os índices (linha, coluna) não nulos e seus valores correspondentes.

Tabela 2: Comparação dos dois corpora

Corpora	Nº de documentos	Nº total de palavras	Palavras únicas (vocabulário)	Nº de palavras por documento (Nº de únicas)	elementos não nulos (em % média)
Movie Reviews	2000	1583820	39768	791,91 (348,37)	0,88%
Base de Santos & Silva	505	8225	2318	16,29 (3,64)	0,16%

Em processamentos que utilizam métodos supervisionados para fazer o aprendizado de máquina do corpora, ou seja, que utilizam uma amostra deste (geralmente 80%) para treinamento do classificador e definição do modelo, é necessário um cuidado especial com o vocabulário criado. Como somente é tratado e considerado no

classificador uma parte do corpora, somente os termos presentes neste grupo de documentos de teste serão adicionados no vocabulário, e, assim, termos novos, ou seja, encontrados nos documentos de teste (partição da corpora não utilizada no treinamento, 20% geralmente) ou em novos documentos fora da corpora, não serão considerados no processamento, podendo ocorrer assim perda de informação e, consequentemente, um erro na classificação. Este erro não ocorre no agrupamento, por exemplo, visto que a cada novo documento adicionado no corpora, todos os documentos são processados, um novo vocabulário é gerado e seus termos ponderados novamente. Soluções são necessárias para resolver este problema, como, por exemplo, obter um dicionário fixo para ser definido como vocabulário que abranja o maior número de termos possíveis ou mais relevantes, e não somente os que existem no grupo de treino. Outra solução é o aprendizado continuo do vocabulário, que exigiria uma nova forma de fazer o aprendizado de máquina em questão.

2.1.4 Ponderação de Termos (TF-IDF)

Como iniciado na seção anterior, diferentes ponderações de termos podem ser utilizadas na representação em *bag-of-words*. O tipo escolhido vai depender do propósito do processamento. Na Tabela 1, por exemplo, foi utilizado como peso dos elementos a frequência pura, ou seja, se a palavra aparece duas vezes no documento, ela terá peso dois neste termo, como é o caso da palavra “gosta” no Ex. 1.

Em alguns estudos, por exemplo, é utilizado o modelo booleano, onde não importa o número de vezes que uma palavra apareceu no documento, somente se ela está presente no mesmo. Este caso pode ser útil, por exemplo, em buscas simples. Métodos estatísticos como o “X^2 statistic”, citado em Miner et al. (2012), também são utilizados, calculando a relação entre as *features*, que são os elementos, e cada rótulo que o processamento se propõe a classificar.

Nesta dissertação foi utilizada a abordagem ponderação de termos baseado em frequências definida pelo método TF-IDF (Liu, 2012), ou *term frequency - inverse document frequency*. Este método foi escolhido pois além de ser um método simples, mas robusto na representação numérica de um texto, é bem utilizado em agrupamento de textos.

Term frequency (TF) é o número de vezes que um termo aparece em um documento. O intuito do TF é valorizar termos que aparecem muitas vezes em um documento, considerando então que eles são relevantes no mesmo. Entretanto, termos muito comuns no corpora podem ser ruins, pois significa que estes não trazem muito valor para o estudo, e não diferenciam os documentos. Além disso, termos que ocorrem pouco podem também ser bastante relevantes, sendo capazes de serem exatamente os que possuem maior significado no documento, servindo para diferenciar e identificar o mesmo ao qual eles estão presentes. Para equilibrar isto existe o IDF, que serve como uma técnica para medir quanta informação um termo trás no conjunto de documentos, visando sanar, assim, um dos problemas do TF, que trata todos os termos como sendo igualmente importantes. O *Inverse Document Frequency* serve justamente para identificar e diminuir o peso de termos muito comuns no corpora, ou seja, que ocorrem muito no conjunto de documentos, e valorizar termos raros, isto é, que ocorrem pouco no corpora. Para explicar melhor como cada técnica é aplicada no tratamento do texto, utilizemos novamente o exemplo dado na seção 2.1.3. Nele pode-se verificar que os termos "gosta" e "de" são termos que possuem frequência "2" nos Ex 1 e 2 respectivamente, apresentando então um TF alto nestes documentos. Entretanto, estes mesmos dois termos estão presentes nos três documentos do corpora, ou seja, não trazem muita informação diferenciada, caso, por exemplo, se queira identificar uma frase específica, como buscar se "João gosta de basquete". Logo, o IDF servirá para reduzir a relevância destes dois termos no BoW, através da sua ponderação (TF-IDF). O termo "de", inclusive, justamente por essa característica de não agregar muito valor no documento, é um dos termos removidos na etapa de *stopwords*, descrita na seção 2.1.2.

O método TF-IDF utiliza essas duas abordagens juntas, multiplicando o TF pelo IDF. Existem diferentes fórmulas para o cálculo desses dois valores, algumas delas levantadas em Manning et al. (2008). Neste estudo o TF corresponde ao número de vezes que o termo t aparece no documento d. Por sua vez, o IDF é representado pela fórmula da Equação 1 a seguir:

$$IDF(t) = \left(log \frac{1+n_d}{1+df(d,t)} + 1\right) \quad (1)$$

Onde n_d representa o número total de documentos do corpora e $df(d,t)$ representa o número de documentos que possuem o termo t. Com o objetivo de se

prevenir a divisão por zero e suavizar o resultado, é somado 1 ao numerador e ao denominador do log do IDF.

Com isso, a formula do TF-IDF resulta na Equação 2:

$$TF \times IDF = tf(t,d) \times idf(t) = tf(t,d) \times \left(log \frac{1+n_d}{1+df(d,t)} + 1\right) \quad (2)$$

Resumindo: caso um termo t apareça muitas vezes num documento d, e esteja presente em poucos documentos do conjunto D, ele possuirá um valor alto de TF-IDF. Entretanto, ele tende a possuir um valor baixo caso esteja presente num grande número de documentos de D.

Após o cálculo do TF-IDF para cada elemento da matriz, foi realizada a normalização de cada linha (cada documento, representado pela letra v) utilizando a distância euclidiana (Eq. (3)):

$$v_{norm} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}} \quad (3)$$

Onde v_i representa o valor de TF-IDF de cada termo único do documento v, $v \in V$, sendo V o conjunto de documentos vetorizados, ou seja, já com seus elementos transformados.

Todos esses cálculos de ponderação de termos e sua normalização, utilizadas nesta metodologia, estão implementadas na biblioteca *TfidfVectorizer*, disponível no Scikit-Learn (TfidfVectorizer, 2017), como melhor descrito na seção 2.1.6.

2.1.5 N-Gram

Na seção 2.1.3 foi falado sobre a *Bag-of-Words*, que é a representação de um documento através dos valores (em função de frequências) de suas palavras. Um problema desse tipo de representação, como já falado, é a perda de aspectos semânticos e de certas informações como: nomes compostos; palavras que juntas possuem mais significado que separadas como, por exemplo, “mineração de texto”, tirando assim possíveis ambiguidades com a extração do minério; e casos que possuem negação, facilitando assim identificar qual palavra ou conjunto de palavras está relacionada a ela.

Na língua inglesa, por exemplo, a diferenciação de duas frases com palavras idênticas, mas com uma delas na forma interrogativa, na qual se inverte a ordem de alguns elementos, a representação em duas palavras juntas, ao invés da palavra sozinha, já ajudaria a diferenciá-las.

O "*Bag-of-Terms*", semelhante ao BoW mas, ao invés de palavras, utiliza termos como seu vocabulário, é uma alternativa visando solucionar esses problemas. Esses termos são definidos pelo que é chamado de modelo N-gram (Liu, 2017). Nele o documento é dividido em *tokens* de tamanho N, isto é, no caso deste estudo, onde trabalha-se com palavras como elemento foco, um modelo 2-gram, ou bigram, divide um documento em conjuntos de 2 palavras, exatamente seguidas uma da outra. Pode-se, inclusive, dizer que o modelo BoW é um caso especial do N-Gram, ou *Bag-of-Terms*, onde N=1, ou seja, é o modelo unigram. Um exemplo desse processamento por bigram de um texto é dado na Figura 3, usando o exemplo 1 da seção 2.1.3. Nele é possível ver que o texto em bigram possui 5 elementos. Cada elemento representará um termo do documento.

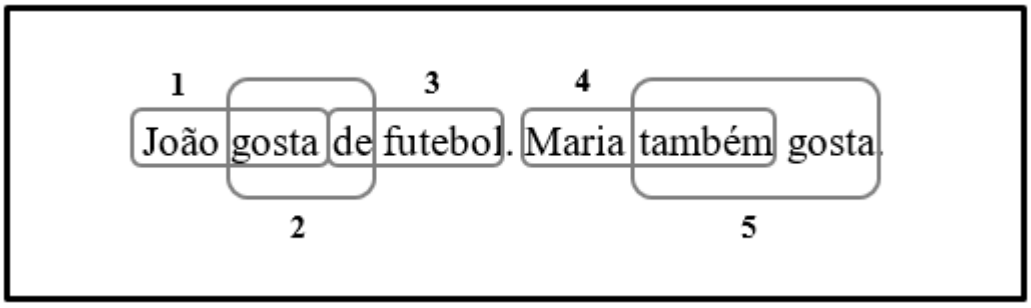

Figura 3: Exemplo de processamento por bigram

Outra forma de abordagem é definir um intervalo de N-gram, como feito por Goyal & Parulekar (2015), ou seja, o Valor de N que será extraído do texto. Utilizando o mesmo exemplo da Figura 3, e considerando um N-gram com intervalo (1-2), seriam extraídos os elementos bigram e os elementos unigram, isto é, os cinco elementos da figura mais os seis elementos unigram, que são as palavras únicas ("gosta" aparece duas vezes).

Outro aspecto importante no uso do N-gram é a possibilidade de identificar padrões, ou seja, conseguir detectar conjuntos de palavras que aparecem juntos com alta frequência. Um exemplo, utilizando o estudo de caso que será descrito mais a frente, é a alta frequência do termo bigram "feto anencéfalo".

Nesta dissertação, uma variável chamada “N-gram” foi definida para representar o número de palavras por termo, ou seja, como os documentos serão quebrados. No exemplo da Figura 3, a variável tem valor 2. Não foi utilizado um intervalo de N-gram, ou seja, se “N-gram” = 2, todos os elementos do vocabulário serão bigram (duas palavras). Esta forma de representação do “N-gram” é chamada N-gram único.

2.1.6 Scikit-learn

O scikit-learn é uma biblioteca código aberto para Python de aprendizado de máquina. Nela estão implementados desde funções de pré-processamento de textos a algoritmos de classificação, agrupamento e regressão. Como exemplo de algoritmos pode-se citar o K-means, SVM, e o Naive Bayes.

Nesta dissertação foi utilizado método “TfidfVectorizer” desta biblioteca para realizar o processamento descrito nas seções 2.1.2, 2.1.3, 2.1.4 e 2.1.5. Este método processa o texto da seguinte forma: através do parâmetro “stop_words” ele recebe a lista do NLTK das mesmas, citada na seção 2.1.2; no parâmetro “ngram_range” é definido qual o intervalo, [valor mínimo, valor máximo], de N-gram que será utilizado para processar os textos; para que a forma de normalização do TF-IDF utilizada seja a euclidiana, o parâmetro “norm” recebe o valor “l2” que corresponde a norma em questão; e são definidas as variáveis “use_idf” e “smooth_idf”, que, respectivamente, definem que o IDF será utilizado no cálculo do TF-IDF e a soma do valor 1 no numerador e denominador do IDF, como descrito na seção 2.1.4. Os outros valores dos parâmetros restantes do método são os já pré-definidos no próprio.

O método citado recebe os documentos, particionados, e processa todas as etapas descritas da seção 2.1.2 a seção 2.1.5, gerando uma matriz esparsa no modelo Bag-of-terms do N-gram definido.

2.2 Agrupamento (Clusterização)

O *K-means* é um método de agrupamento no qual os elementos são reagrupados a cada iteração, recalculando o centroide até atingir o equilíbrio dos elementos em cada grupo. Não foi considerado nenhum valor inicial para a semente do algoritmo. Ao invés disso, foi utilizado o método *K-means++*, sugerido por Arthur & Vassilvitskii (2007) e

implementado no scikit-learn, que modifica a forma de inicialização dos centroides. No algoritmo padrão os K centroides iniciais são escolhidos de forma randômica ou são definidos na entrada do algoritmo pelo usuário. Desta forma, o algoritmo converge após um tempo de execução suficiente. Porém, ele pode convergir para mínimos locais, devido à proximidade das sementes. O método *K-means++* tenta minimizar esse problema, definindo centroides geralmente mais distantes entre si. Este algoritmo é apresentado abaixo:

1. Escolha dos centroides iniciais:
    a. Define-se um centroide c_1, escolhido uniformemente de forma aleatória de X.
    b. Define-se um novo centroide c_i, escolhendo $x \in X$ com probabilidade $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$.
    c. Repete-se o passo 1.b até se ter todos os k centroides.
2. Passos normais do K-means:
    a. Para cada $i \in \{1, \ldots, k\}$, o cluster C_i é definido como o conjunto de elementos (pontos) de X que são mais próximos de c_i do que c_j, para todo $j \neq i$.
    b. Para cada $i \in \{1, \ldots, k\}$, define-se como c_i o centro de massa de todos os pontos em C_i: $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.
    c. Repete-se os passos 2.a e 2.b até C não se modificar mais.

Os outros elementos deste método não foram alterados, com exceção do "random_state", que possui por padrão o valor "None", mas foi definido com "0" para este estudo. Este valor é utilizado como valor inicial da geração de números aleatórios do método.

2.2.1 Combinação de parâmetros

Como foram definidas nas seções anteriores, existem três varáveis principais nesta metodologia: "Textual", referente ao particionamento do documento; "N-gram", representando o número de palavras do termo, e "K", o número de clusters (grupos) utilizados. Em uma execução da metodologia estas três variáveis precisam ser definidas. Por exemplo, considere o cenário de um conjunto de documentos, estes possuindo duas partições, ou seja, três valores possíveis para a variável "Textual" (0, 1 ou 2), cujos termos

são formados por quatro palavras. Caso se queira agrupar estes documentos em três clusters, utilizando somente a primeira parte de cada documento (“Textual 1”), a execução será definida como: K= 3;Textual=1;N-gram=4.

Inicialmente, buscou-se identificar da combinação de valores dessas variáveis aquela que apresentasse os melhores resultados, ou seja, que identificasse e separasse de forma mais correta os documentos de acordo com o objetivo do estudo de caso. Porém, como melhor descrito na seção 3.2 do estudo de caso, não é uma tarefa tão trivial e nem eficaz por si só. Logo, buscou-se fazer uma combinação dessas execuções para analisar o número de vezes que os documentos aparecem num mesmo agrupamento, entender suas semelhanças e diferenças e extrair resultados mais relevantes para o estudo.

Cada uma destas variáveis possui um intervalo, ou seja, qual o menor e maior valor que cada uma utilizará para execução, com estes dois limites inclusos no intervalo, ou seja sendo executados. Por exemplo, caso se queira testar a metodologia agrupando os documentos em três grupos e, posteriormente, em 4 e 5 grupos, “K” possuirá um intervalo de 3 a 5, representado na forma K = (3:5). A mesma representação vale para as outras duas variáveis. Desta forma, cada valor de cada intervalo dessas variáveis será combinado e executado, e é chamado de Combinação de parâmetros ou Combinação de execução. Caso se tenha, por exemplo, uma combinação, chamada de E1, do tipo: K=(2:3); Textual(0:1);N-gram=(1:3), cada valor de cada variável será combinado e uma execução será gerada, com seus resultados analisados de forma conjunta. A Tabela 3 representa todas as execuções do exemplo citado, ou seja, 12.

A Figura 4 representa a generalização desta combinação de parâmetros, onde k_1 e k_2 representam os valores mínimos e máximos de grupos, t_1 e t_2 os mínimos e máximos dos valores de textual a serem executados e n_1 e n_2 o número mínimo e máximo de palavras por termo. Essa combinação dos três parâmetros gera M execuções (combinações).

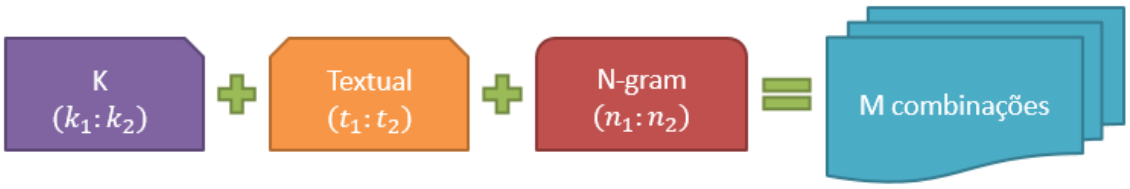

Figura 4: Combinação de parâmetros

Tabela 3: Combinação E1{K=(2:3); Textual(0:1);N-gram=(1:3)}

Nº execução	K	Textual	N-gram
1	2	0	1
2	2	0	2
3	2	0	3
4	2	1	1
5	2	1	2
6	2	1	3
7	3	0	1
8	3	0	2
9	3	0	3
10	3	1	1
11	3	1	2
12	3	1	3

Utilizemos a combinação E2 = K=(2:3); Textual(0:1);N-gram=(1:1), ou seja, 4 execuções diferentes, e um corpora contendo 6 documentos, para exemplificar como os resultados são combinados e como essa combinação de parâmetros é utilizada e visualizada. Supõe-se também que os documentos possuem como objetivo de estudo 2 classificações possíveis (para facilitar o exemplo chamadas de positivo ou negativo), que são seus rótulos. E2 gera 4 clusterizações dos documentos, representadas na Figura 5. Os documentos verdes representam os positivos e os vermelhos os negativos.

O primeiro passo da análise de resultado de cada combinação é a contagem de quantos positivos e quantos negativos cada cluster possui. Estes valores são apresentados na Tabela 4. Como visto na tabela, conseguir uma execução que consiga separar exatamente as classificações objetivo, ou, pelo menos, de forma mais eficiente possível, é uma tarefa muito difícil. Porém, é interessante, por exemplo, analisar os comportamentos dos documentos na clusterização, ou seja, certos documentos que costumam aparecer juntos em um grupo. O número do cluster (1, 2 ou 3) não importa nesta análise e sim a formação do grupo em si. Esta mudança de avaliação dos resultados e estudo das execuções é melhor descrito na seção 3.2. Assim, será possível, por exemplo, desambiguar grupos que não possuem uma classificação clara (cluster 1 da execução 1, por exemplo), entender o porquê desse comportamento e, com isso, utilizar outras

técnicas para agrupar clusters com predominância de classes iguais, isto é, juntar os grupos que possuem maioria positiva em um só.

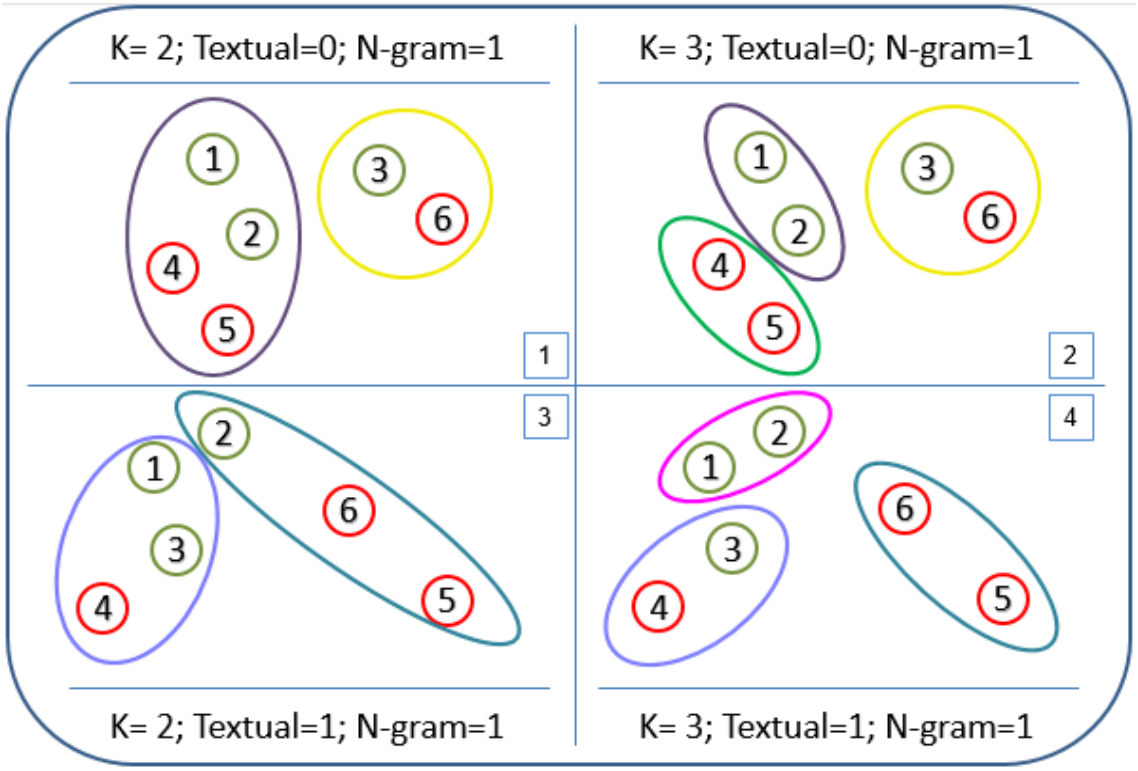

Figura 5: Clusterizações geradas pela combinação E2

Tabela 4: Contagem de positivos e negativos da combinação E2

Execução	Cluster 1	Cluster 2	Cluster 3
1	2 pos. / 2 neg.	1 pos. / 1 neg.	--
2	2 pos. / 0 neg.	1 pos. / 1 neg.	0 pos. / 2 neg.
3	2 pos. / 1 neg.	1 pos. / 2 neg.	--
4	1 pos. / 1 neg.	0 pos. / 2 neg.	2 pos. / 0 neg.

Entretanto, analisar cada execução manualmente, uma a uma, para identificar padrões de comportamento e documentos que costumam aparecer juntos num mesmo cluster é uma tarefa praticamente impossível, ou demandará um tempo muito grande, se for considerado que geralmente o número de combinações costuma ser bastante elevado. Viu-se necessário o desenvolvimento de novas técnicas para se visualizar esses comportamentos.

2.3 Visualização em Grafos

Com objetivo de visualizar melhor estas ligações recorrentes entre documentos, ou seja, documentos que frequentemente estão juntos em um certo cluster, foi proposta a representação em grafos.

Os vértices dos grafos representam os documentos e as arestas significam que dois deles se encontram em um mesmo grupo em uma execução. Por exemplo, nas duas primeiras execuções da combinação E2, ou seja, {K=(2:3); Textual(0);N-gram=(1:1)}, um grafo como o representado na Figura 6 é gerado. Os documentos 1 e 2 aparecem em um mesmo grupo nas duas execuções, logo possuem duas ligações (arestas) entre si.

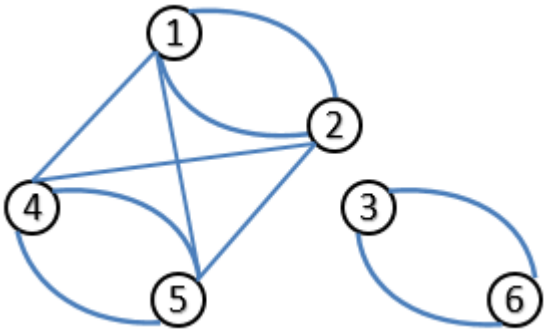

Figura 6: Grafo gerado da combinação anterior

A geração e representação em grafos das combinações foi feita utilizando a biblioteca igraph (igraph, 2017), também da linguagem Python.

2.3.1 Coloração de vértices

Numa segunda etapa, a fim de se melhorar a visualização e facilitar observações sobre as ligações e padrões, foi feita a coloração dos vértices. Foi testado inicialmente a coloração de acordo com os clusters a que eles pertencem. Como em diferentes execuções documentos podem aparecer em diferentes clusters, é necessário escolher qual a execução que será utilizada como base para a coloração. A Figura 7, nos itens (a) e (b), ilustra a coloração do grafo da Figura 6 em relação às duas execuções que o grafo representa, com K = 2 e K = 3, respectivamente. Na seção 3 é descrito com mais detalhes o porquê deste tipo de coloração não ser tão eficiente. Mas, basicamente, esta forma não traz muita informação, visto que o cluster especifico ao qual os documentos pertencem não é tão

importante e que analisar a mudança de coloração, execução por execução, ainda é um trabalho muito grande para ser realizado manualmente.

Uma segunda solução, com o objetivo de agregar mais valor ao grafo e à análise dos seus resultados, é colorir os documentos de acordo com os rótulos (classificações) pré-definidos. Esta nova coloração do grafo está representada no item (c) da Figura 7.

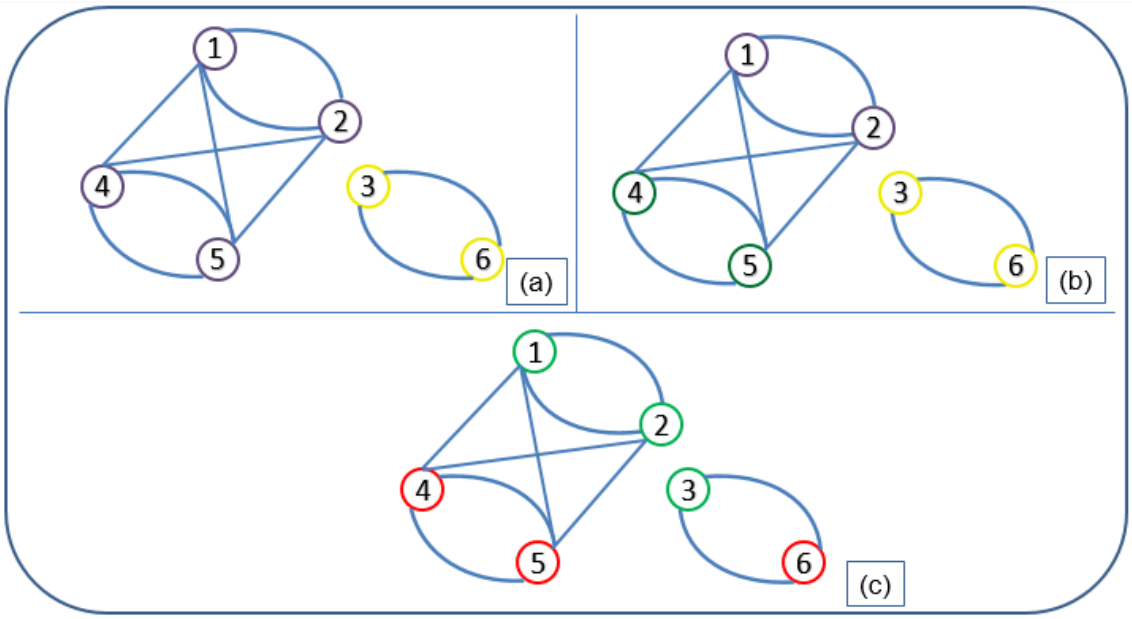

Figura 7: Coloração de vértices de acordo com o agrupamento (a) e (b) e de acordo com a polaridade (c)

Esta nova forma permite analisar melhor os resultados e o avanço da metodologia, identificando se projetos de mesmo rótulo possuem maiores ligações ou casos onde projetos com rótulos diferentes possuem ligações mais forte. Os vértices 3 e 6 do item (c) da Figura 7 exemplificam este último caso citado. Analisar e entender o porquê deste comportamento é um dos objetivos desta dissertação. Análises mais profundas sobre estes casos são feitas na seção 3.2 e no capítulo 4.

2.3.2 Ponderação de arestas

Em um número pequeno de execuções, como no exemplo da Figura 7, visualizar perfeitamente todas as ligações do grafo é uma tarefa bem simples. Mas como será visto no estudo de caso, o número de combinações utilizadas é bem alto, e com isso o grafo tende a se tornar mais complexo de se visualizar. Por exemplo, em um processamento de

50 ligações, imaginemos que em 45 delas dois documentos estão no mesmo grupo. Ou seja, entre dois vértices haverá 45 arestas. Considerando um grupo de 20 documentos com comportamentos similares, visualizar e entender um grafo destes será muito difícil. Na seção 3 estes problemas ficam mais claros.

Assim, a ponderação de arestas ajuda a diminuir essa "poluição" do grafo, reduzindo todas as arestas entre os mesmos dois vértices A e B, por exemplo, em uma só, e aplicando um peso a ela, referente ao número de arestas existentes neste intervalo. O item (a) da Figura 8 aplica a ponderação de arestas no grafo da Figura 7. A fim de se destacar vértices que possuem mais ligações entre si, a ponderação foi posteriormente representada pela espessura da aresta, ou seja, quanto maior o número de ligações, maior a espessura. O item (b) da Figura 8 ilustra essa nova representação.

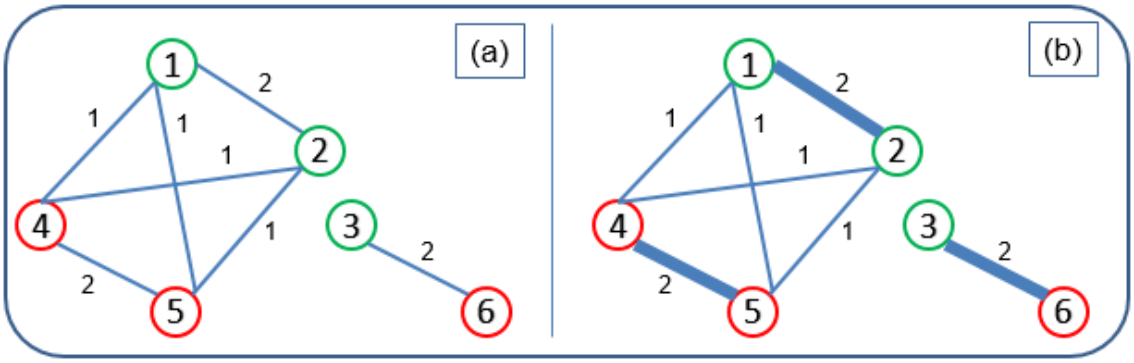

Figura 8: Grafo com arestas ponderadas

Como são analisados casos nos quais se utiliza tanto duas execuções como 50, uma forma de se representar e padronizar essa espessura era necessária, pois utilizar o valor exato da ponderação, em pixels ou de forma proporcional, para a espessura da aresta trouxe problemas, como visto na Figura 18 da seção 3.2.1. A solução encontrada foi definir valores mínimo e máximo para a espessura da aresta, sendo o valor mínimo referente ao menor número de ligações entre dois vértices da execução representada pelo grafo, e o valor máximo o número máximo de ligações. O resto dos valores dentro deste intervalo são normalizados dentro destes limites. A Figura 19 da seção 3.2.1 ilustra o resultado desta normalização.

2.3.3 Top Grafos

Mesmo com a ponderação de arestas, esta redução de complexidade do grafo ainda não é suficiente para que seja possível ter uma análise mais profunda do mesmo, como melhor demonstrado na seção 3.2.1. Grafos que representam um grande número de combinações de execução, como é o caso do estudo de caso desta dissertação, ainda são muito "poluídos" visualmente. Como o objetivo principal deste estudo é identificar e entender as relações entre documentos que possuem alta similaridade e características comuns, acarretando, assim, num alto número de ligações (arestas) entre eles, foi levantada a técnica de Top grafo.

O top grafo visa destacar os documentos que possuem as maiores ligações entre si do grafo incialmente gerado, que passaremos a chamar de "grafo normal". Para a geração deste novo grafo, é definido um "valor de corte". Este valor define, em porcentagem, o número mínimo de ligações (peso das arestas) necessárias para que os vértices que formam o par através destas arestas sejam visualizados no top grafo. Um top grafo com valor de corte de 100% possuirá somente vértices cujas arestas possuem valor máximo, ou seja, documentos que estão juntos no maior número de grupos possível (não necessariamente em todos as execuções, mas no maior número alcançado na combinação de execução). Em contrapartida, um corte 0% apresentará vértices que possuem pelo menos uma ligação, ou seja, o top grafo com corte 0% representa exatamente o grafo normal por inteiro. A Figura 9 representa o top grafo com corte 100% do grafo da Figura 8, ou seja, somente vértices cujas arestas possuem valor máximo, que neste caso é 2. Neste exemplo todos os vértices do grafo normal estão presentes também neste top grafo, mas com execuções maiores, o número de vértices tende a diminuir, principalmente no corte 100%, tornando a visualização mais simples de ser analisada.

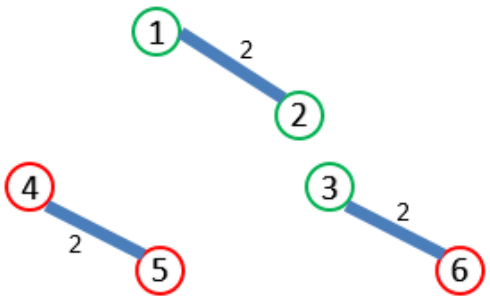

Figura 9: Grafo com corte de 100%

Capítulo 3

3.Estudo de caso

Para este estudo 33 projetos de lei brasileiros (PLs) foram fornecidos por uma especialista do Programa de Pós-Graduação de Ciência Política da USP, cujo estudo é focado no tema aborto e projetos de lei que o abordam. Juntamente aos projetos foi fornecida uma tabela, semelhante à mostrada na Tabela 5, representando a polaridade de cada documento, rotulados de forma manual pela própria especialista. Na Tabela 6 são apresentadas as quantidades de PLs presentes em cada polaridade.

Tabela 5: Amostra de projetos de lei rotulados

Projetos de lei	A favor ou contra o aborto
PL4834_2005	A favor
PL5364_2005	A favor
PL478_2007	Contra
PL489_2007	Contra
PL660_2007	A favor

Tabela 6: Quantidade e polaridade rotulada dos projetos de lei

Polaridade	Número de projetos de lei
A favor do aborto	13
Contra o aborto	20

3.1 Pré-processamento

Inicialmente, os projetos de lei foram analisados manualmente, um a um, para se verificar sua estrutura. Os PLs seguem certo padrão com relação a sua apresentação,

podendo assim ser dividido em três partes: uma breve descrição sobre o assunto que a lei tratará; o "Decreto" em si, que enuncia a lei a ser proposta ou uma alteração de lei já vigente; e, por último, uma "Justificativa", contendo os argumentos apresentados por meio de um texto corrido em favor da proposta apresentada. Um exemplo de projeto de lei e suas subdivisões é demonstrado na Figura 10. O início da "Justificativa" no PL é definido por um subtítulo.

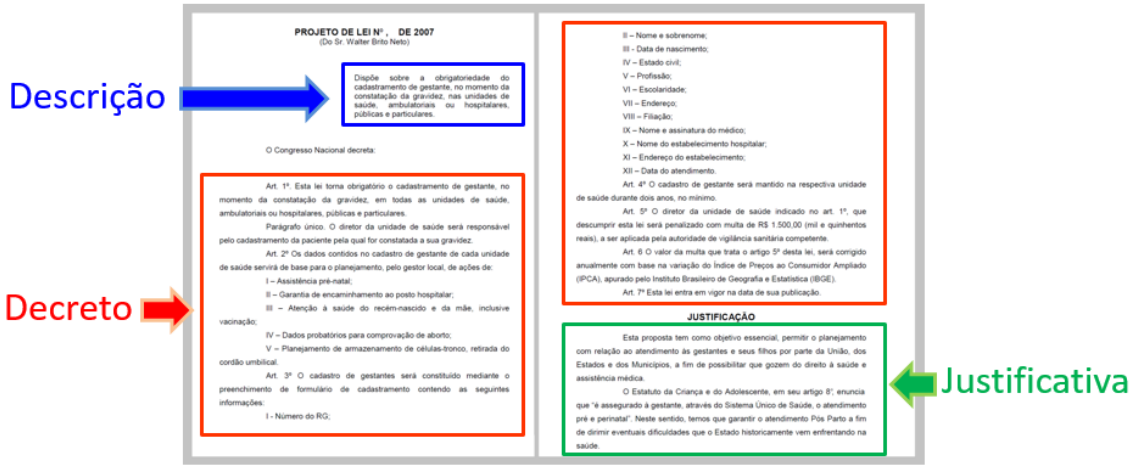

Figura 10: Exemplo de projeto de lei

Entretanto, eles não seguem fielmente estas definições de forma tão padronizada, dificultando um pouco o processamento textual. Por exemplo: Alguns projetos se referem à "Justificativa" como "Justificação" em seu subtítulo; outros omitem o subtítulo do tópico, diferenciando do texto do decreto apenas pela quebra de página; alguns outros modificam a forma como a palavra é representada, incluindo espaços entre cada letra ("J U S T I F I C A Ç Ã O"). Considerando os 33 exemplares adquiridos para este estudo, contornar este problema é relativamente simples. Visando trabalhos futuros, em que um montante bem maior de projetos poderá ser tratado, essa falta de padronização na estrutura do projeto pode se tornar uma dificuldade relevante.

Outra dificuldade identificada ainda no levantamento de dados a serem analisados foi com relação aos decretos. Alguns apenas modificam uma lei, removendo um inciso, ou até revogando uma lei inteira. Nestes casos, estes projetos acabam não possuindo informações relevantes na análise textual, dificultando seu processamento para identificação de polaridade. Justificações muito pequenas ou sem relevância também representam um empecilho neste processamento. Quanto menos informação mais difícil

se torna a tarefa de polarização, principalmente na justificação, onde o texto é menos imparcial e onde o autor pode argumentar e demonstrar suas ideias e convicções.

Uma prática muito comum na Câmara é a apensação de projetos. Como definido pelo próprio site da Câmara dos deputados do Brasil (Câmara, 2017), projetos apensados são projetos de conteúdo semelhante que tramitam em conjunto. Apesar de continuarem possuindo suas individualidades como seus próprios nomes, são processados pelo legislativo de forma conjunta, com parecer único do relator, apesar deste ter que se pronunciar sobre todos os apensados. Esta prática de apensar projetos também é levantada na seção de trabalhos futuros (5.1) como uma possível solução do problema de justificativas pequenas ou pouco informativas. Além disso, esta metodologia desenvolvida pode servir de ferramenta para automação deste processo de apensação. A Figura 11 exemplifica como um projeto é referenciado ao projeto principal que representa os apensados, item (a) (PL 5364/2005, 2017), e como a lista de apensados é representada no site da Câmara, item (b) (Apensados, 2017).

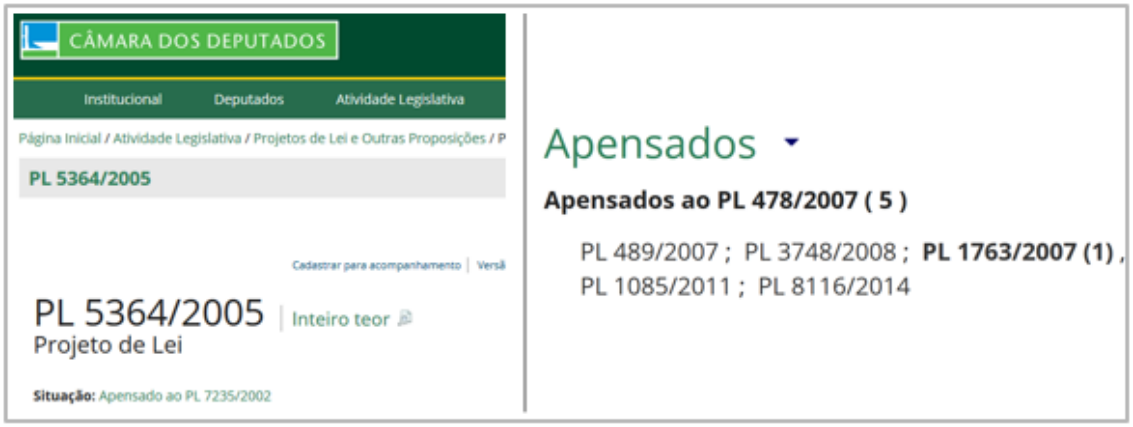

Figura 11: Projeto apensado

Os PLs foram processados automaticamente e salvos em memória de forma estruturada, separando o "decreto" e a "justificativa", e ordenados pelo Ano e nome, respectivamente. Para isto, foram utilizados os padrões identificados manualmente e suas exceções, já exemplificadas anteriormente. Todas as palavras foram reduzidas a forma caixa baixa (letras minúsculas).

Neste estudo, somente o "decreto" e a "justificativa" foram processados, tanto juntos como de maneira individual, para que se avalie o impacto de cada um e o quão cada parte é mais significativa para a polarização. Desta forma, essas três possibilidades

de texto são utilizadas como descrito na Tabela 7, dependendo do valor da variável "Textual". Por exemplo, se "Textual" for 1, somente o decreto é processado e analisado.

Tabela 7: Valores da variável "Textual"

Valor da variável "Textual"	O que representa
0	O Projeto todo – decreto e justificativa
1	Somente o decreto
2	Somente a justificativa

Foi escolhido não tratar a primeira parte do PL, que apresenta uma breve descrição do documento, por ser muito vaga e pequena (poucas palavras), não sendo assim muito relevante para o processamento aqui feito e podendo gerar erros e até atrapalhar o processo devido ao padrão de palavras que poderia gerar.

Em seguida foram removidas as *stopwords* e os PLs foram representados através de *bag-of-Terms* com seus elementos sendo ponderados utilizando a técnica TF-IDF definida na seção 2.1.4, com termos no formato N-gram único, onde "N-gram" varia de 1 a 4 palavras, gerando assim, uma matriz PL x Termo. Além das variáveis "Textual" e "N-gram", o intervalo de "K" foi testado de 2 a 7.

Antes de definir este intervalo do N-gram, a fim de se analisar as principais palavras e expressões utilizadas, a tabela PLs x termos foi processada, extraindo assim os maiores pesos para cada termo (sendo o termo um N-gram) e ordenando por maiores valores, gerando uma lista dos 300 termos q possuem maiores pesos da base de dados. Foi elaborada uma lista para cada N-gram,, testando valores de 1 a 5, ou seja, 6 listas no total. Esta análise foi importante em alguns aspectos: a partir dela, com a ajuda da especialista, foi possível verificar que o n-gram igual a 5 não trazia muita informação e não se mostrou muito relevante na pesquisa, por esta razão foi definido o intervalo somente de 1 a 4; foram verificadas palavras comuns deste tipo de documento que não trazem informação importante que identifiquem ou tragam relevância ao texto. Estas palavras, futuramente poderiam ser testadas adicionando-as à lista de *stopwords*. Este teste ficou para trabalhos futuros.

Além disto, através destas listas geradas, foi possível identificar expressões e palavras muito usadas neste contexto (leis sobre aborto) e, assim, como melhor

desenvolvido na seção 5.1, elas podem ser uteis na desambiguação de grupos e ajudar na polarização de documentos. Palavras como “nasciturо”, por exemplo, geralmente são usadas por pessoas com vieses contra a liberalização do aborto, logo tende a aparecer somente em textos contrários.

Nesta análise, foram combinadas as três variáveis citadas anteriormente: “Textual”, “N-gram” e “K”, gerando assim um total de 72 combinações de execução. A Figura 12 representa esta combinação.

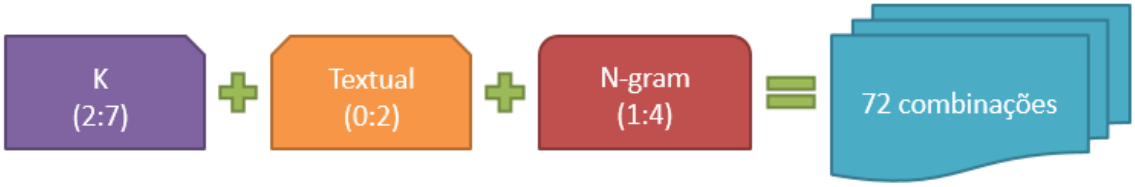

Figura 12: Combinações realizadas no estudo de caso

3.2 Visualização de resultados

Uma primeira avaliação foi a análise de quantos projetos negativos e positivos cada grupo de cada execução apresentava, no qual, por exemplo, os favoráveis e contrários se isolassem em seus grupos. Na Figura 13 é exemplificada a combinação {K=5; Textual=0; N-gram=4}, chamada de combinação C1.

Os elementos vermelhos representam os projetos contra o aborto enquanto os verdes a favor. Nela é possível observar que o método separa perfeitamente os projetos nos grupos 0, 1 e 4. No grupo 2 é verificado somente um PL discrepante do resto de seus elementos. O grupo 3, entretanto, que contém a maior quantidade de projetos, sendo nove contra e oito a favor, possui uma certa similaridade entre seus termos. Foi verificado que na maioria das combinações de parâmetros, no mínimo um agrupamento apresentava este mesmo comportamento.

Um aspecto interessante é analisar no decorrer de várias execuções a ocorrência de PLs juntos em um mesmo grupo, independentemente de sua polaridade pré-definida. Estes projetos podem apresentar características semelhantes que os coloquem juntos em diversas execuções do algoritmo com parâmetros distintos. Entender o porquê desse comportamento se tornou um dos focos deste estudo.

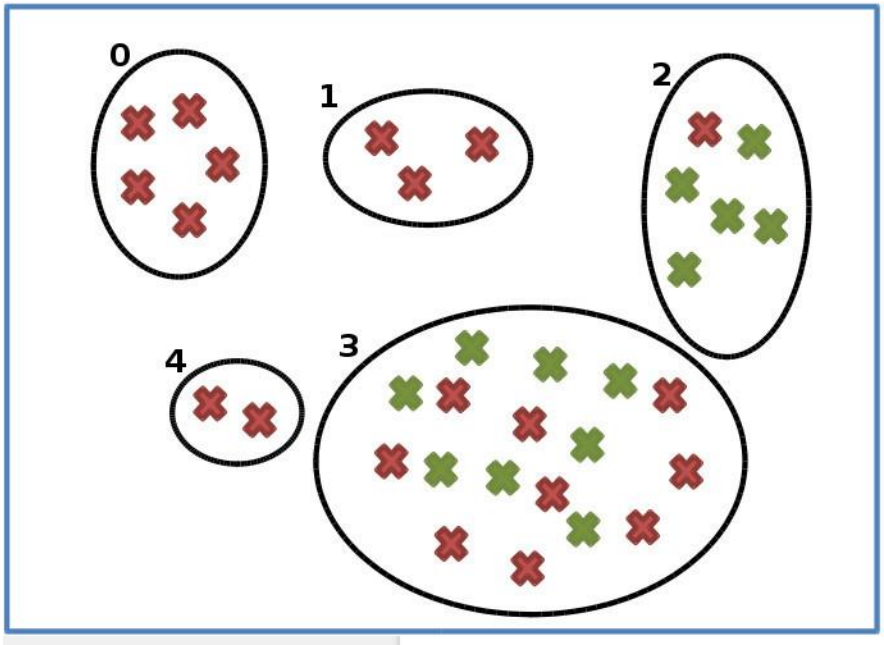

Figura 13: Análise de cenário da combinação C1

O elevado número de combinações dificulta a análise individual dos gráficos ou dados em busca das similaridades entre os PLs. Foram desenvolvidas outras formas de representação destes resultados, como já descritos na seção de metodologia.

3.2.1 Visualização em grafos

Com objetivo de visualizar melhor estas ligações recorrentes entre PLs, foi proposta a representação em grafos, utilizando a biblioteca igraph em Python (igraph, 2017). Os vértices dos grafos representam os PLs e as arestas significam que dois PLs se encontram em um mesmo grupo em, pelo menos, uma combinação de parâmetros.

Por exemplo, no grupo 1 da combinação C1, demonstrada na Figura 13, cada PL possui 2 arestas, ligando-se aos outros 2 elementos do grupo. Na Figura 14 é apresentado um exemplo de grafo, correspondente à combinação C2 {K=(5:6), Textual=0; N-gram=4}, onde "K=(5:6)" significa que o grafo representa as execuções com "K"=5 e "K"=6. A coloração dos vértices inicialmente foi feita, como descrita na seção 2.3.1, em relação ao cluster ao qual o PL está presente na execução com "K"=5. A Figura 15 representa o mesmo grafo, mas com coloração relativa aos clusters da execução "K"=6.

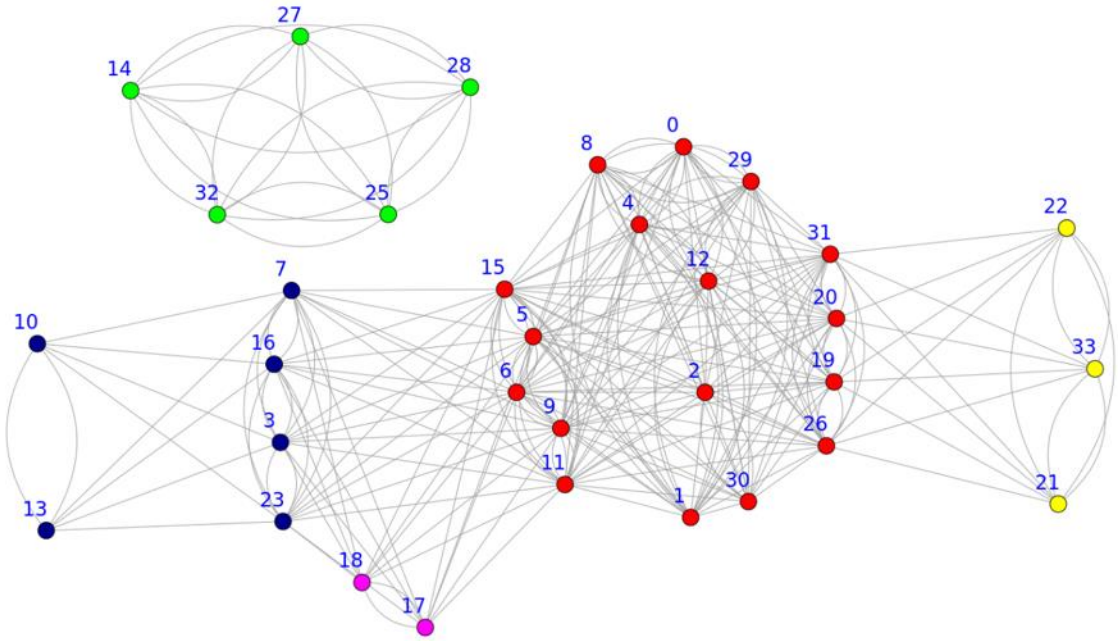

Figura 14: Grafo da combinação C2 com coloração em relação ao cluster K=5

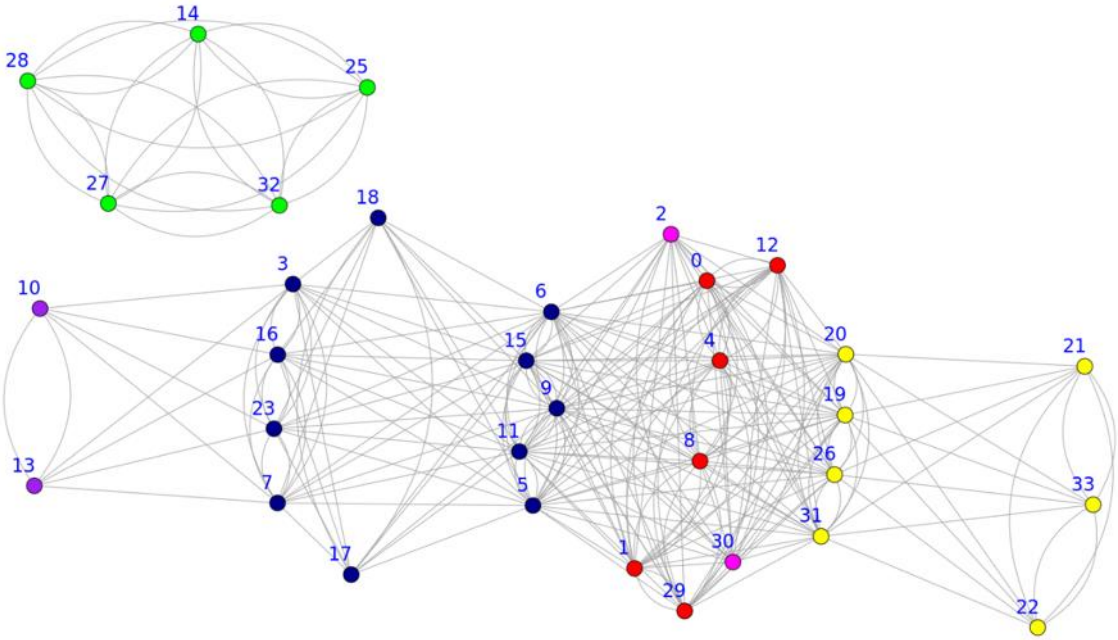

Figura 15: Grafo da combinação C2 com coloração em relação ao cluster K=6

Apesar de se identificar aspectos interessantes como, por exemplo, a não mudança de cluster dos conjuntos de PLs [14, 25, 27, 28, 32] e [21, 22, 33], além de projetos como o 10 e o 13 que apesar de terem mudado de cluster, continuam juntos, essa forma de coloração, como já iniciado na seção 2.3.1, não ajuda muito na melhor analise do grafo.

Assim, optou-se pela coloração utilizando os rótulos pré-definidos de cada PL. O grafo resultante dessa nova representação está representado na *Figura 16.*

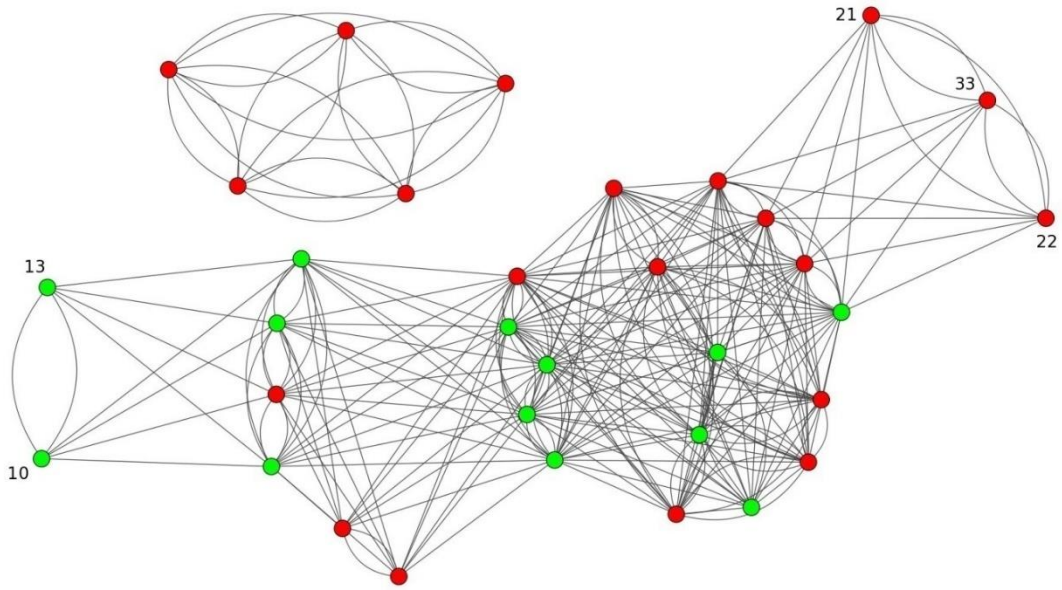

Figura 16: Representação em grafo da combinação C2

A cor dos vértices representa o rótulo de polaridade de cada projeto, fornecido pela especialista. Projetos contra o aborto receberam a cor vermelha e a favor receberam a cor verde. Esta solução trouxe um avanço na análise em alguns aspectos, tal como a similaridade de comportamento e destaque de alguns PLs. Os PLs 13 e 10 apresentam um comportamento semelhante, ligando-se entre si e aos mesmos vértices. O mesmo vale para os PLs 21, 22, 33. Estas ligações foram relatadas à especialista, que observou que os PLs (10 e 13) e (21 e 33) tratam-se de projetos equivalentes entre si. “Projetos equivalentes” são projetos que foram reescritos, ou submetidos novamente para apreciação, utilizando a mesma proposição de lei e na maioria das vezes a mesma justificativa ou uma similar, reforçando a anterior. Ou seja, estes projetos possuem praticamente o mesmo conteúdo, mudando somente o seu identificador e as palavras utilizadas.

Na Figura 17 é destacado o subgrafo isolado do grafo da Figura 16 com os respectivos nomes e índices dos PLs. Estes 5 PLs tiveram apenas ligações entre si. Outra observação relevante neste subgrafo é que todos possuem rótulos contra o aborto. A partir de grupos como este é possível analisar detalhes comuns e assim utilizá-los para identificar grupos de mesma polaridade. Uma análise mais detalhada dessas ligações é discutida na seção 4.

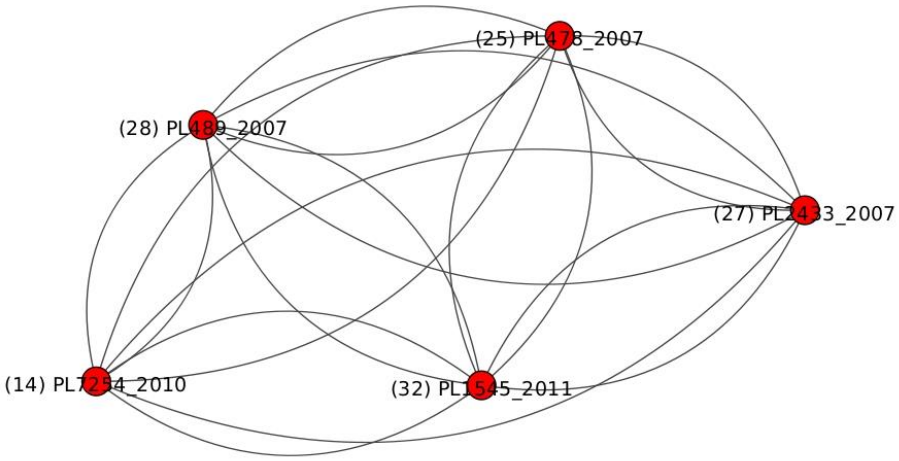

Figura 17: Destaque do grupo independente da Figura 16

O problema desta forma de visualização, como já citado na seção 2.3.2 é o alto número de arestas, que dificulta a verificação de algo relevante de forma manual. Enquanto informações relevantes podem ser levantadas de grupos em destaque, como os três citados anteriormente, a análise do resto do grafo se torna inviável manualmente, como é o caso do grande "emaranhado" de arestas na parte central do grafo da Figura 16. Uma solução levantada nessa seção foi a ponderação de arestas, resumindo todas as existentes entre os mesmos dois vértices A e B numa só, com o número de arestas representadas nela como peso. A espessura desta aresta única é proporcional ao número de arestas que ela representa. Uma combinação aleatória possuindo seis execuções foi gerada somente para se ter uma ideia do resultado dessa ponderação, ilustrada na Figura 18.

Como visto na Figura 18, arestas com valores muito baixo de ponderação acabam se tornando difíceis de serem visualizadas. Por esta razão foi desenvolvido uma forma de normalização de arestas, descrita na seção 2.3.2. O resultado desta normalização é ilustrado na nova representação do grafo da Figura 18 na Figura 19.

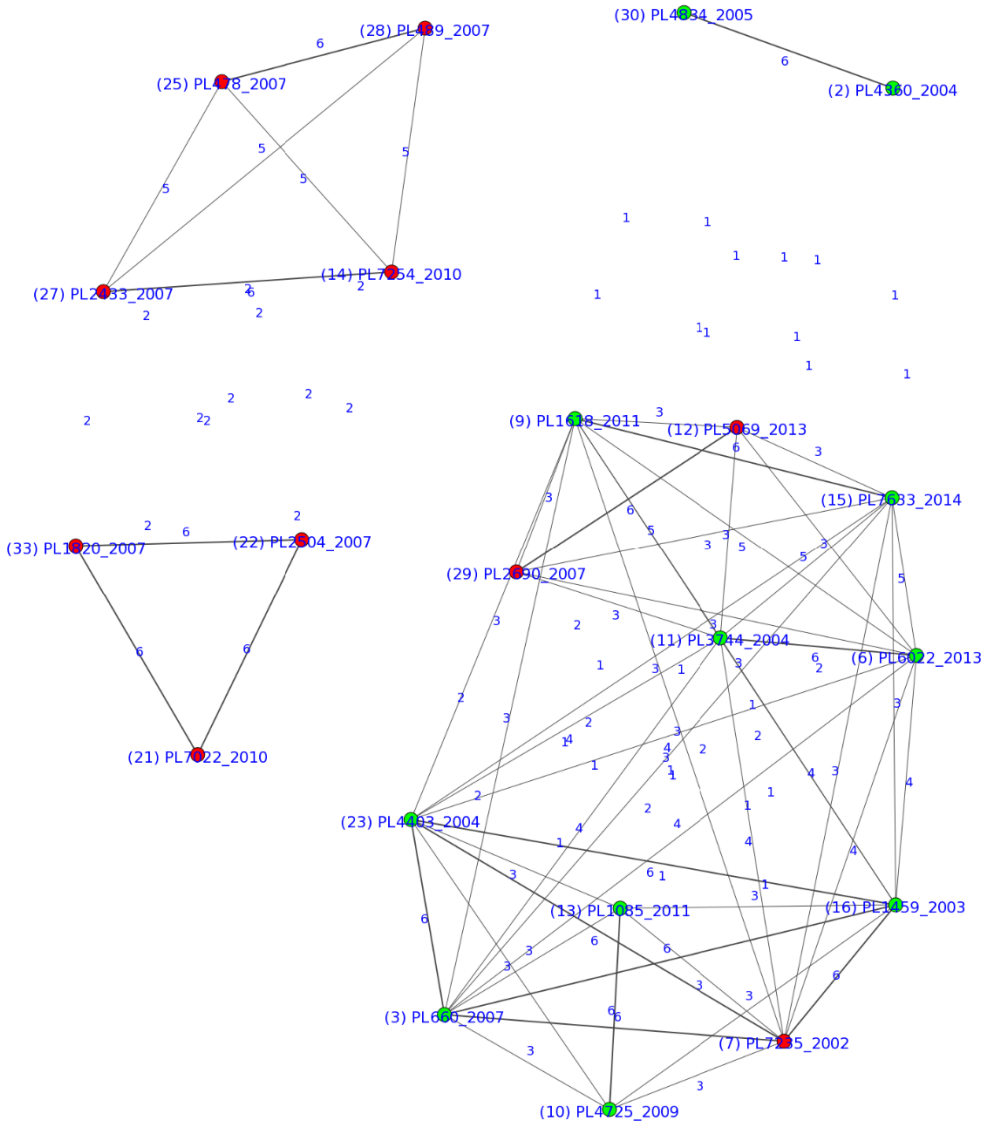

Figura 18: Exemplo de combinação de seis execuções com arestas ponderadas

A Figura 20 apresenta o mesmo grafo presente na Figura 16 mas com suas arestas ponderadas e a suas espessuras normalizadas. É possível perceber que a visualização já fica um pouco mais clara, podendo identificar melhor projetos que possuem mais ligações. Entretanto esta solução ainda não é suficiente para tornar a análise manual mais viável, ainda está confusa em algumas partes, como no centro do grafo, principalmente considerando que esta combinação representada possui um baixo número de execuções.

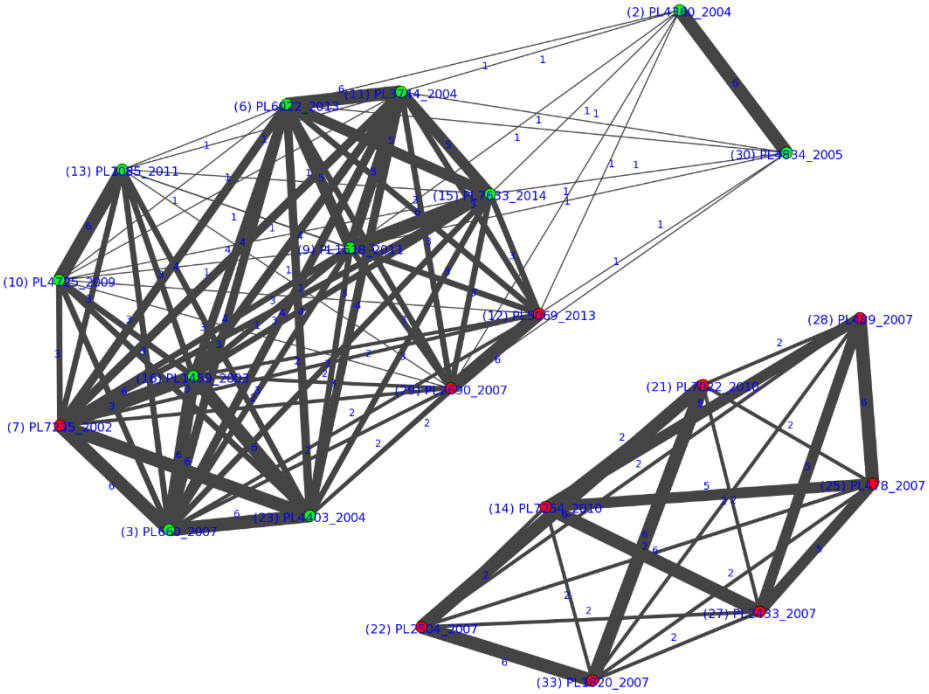

Figura 19: Normalização das arestas da combinação de seis execuções

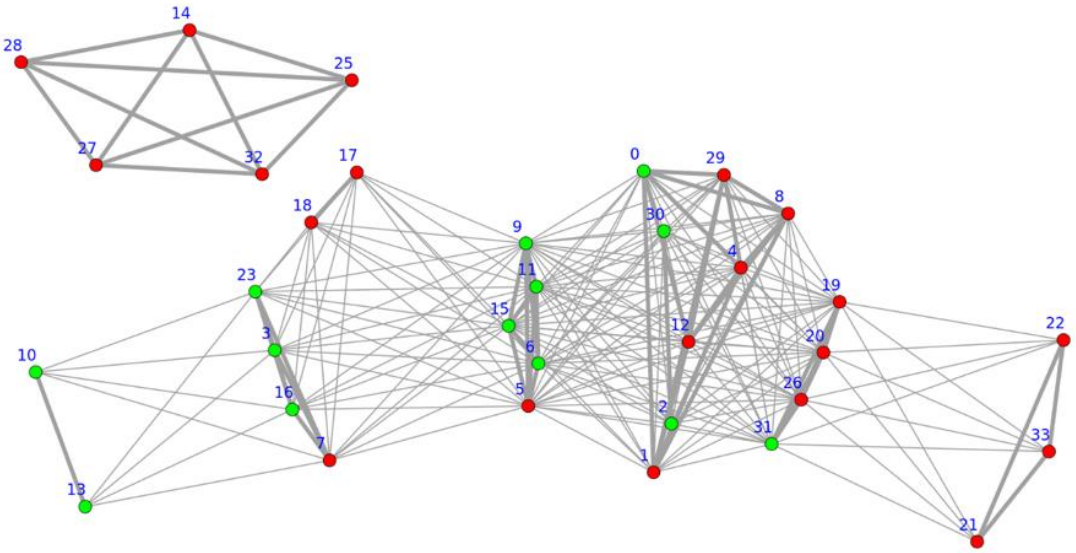

Figura 20: Arestas ponderadas da Figura 16

Para uma avaliação mais abrangente de todas as ligações entre PLs computadas, foi gerado um grafo contendo as 72 combinações de execução. Um exemplo da dificuldade citada é mostrado na Figura 21 que representam estas 72 combinações de

execução, ou seja, todas as execuções, com “K” variando de 2 a 7, “N-gram” de 1 a 4 e “Textual” de 0 a 2, chamada de combinação C3. Nela é possível ter noção da dificuldade

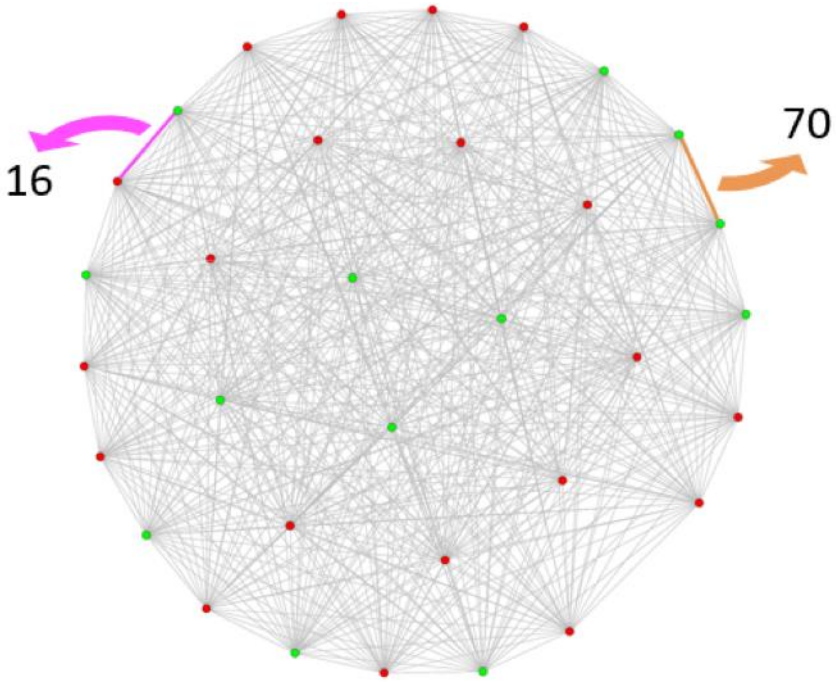

Figura 21: Representação em grafo da combinação C3

de se visualizar o grafo a fim de se tirar conclusões sobre suas ligações, mesmo com as arestas ponderadas.

Como definido na seção 2.3.3, os top grafos foram gerados para tentar sanar esse problema de visualização, e assim melhorar a observação dos resultados, focando nas maiores ligações entre projetos. Nestes grafos as ponderações com valor abaixo do corte definido são eliminadas. Por exemplo, considerando a combinação C3 {K=(2-7);Textual=(0-2);N-gram=(1-4)} com um corte de 100%, o grafo somente apresenta os vértices que possuem as ligações com o valor máximo (72), presente na Figura 22. Um corte de 0% significa que o grafo não cortará nenhuma aresta, ou seja o grafo será igual ao grafo completo representado na Figura 21. Já é possível identificar neste corte de 100% uma vantagem desse método, pois todos os pares apresentados possuem mesma polaridade, demonstrando que projetos que estão presentes em todas as execuções são de alguma forma semelhantes.

Na combinação C3 com o corte de 80% (57 ligações ou mais), apresentada na Figura 23 pode ser observado que todas as ligações de PLs possuem mesma polaridade. Por outro lado, para a combinação C3 com corte de 70% (50), Figura 24, algumas ligações

entre PLs com polaridades opostas ocorrem, junto com casos de ligações entre PLs semelhantes. As análises destes comportamentos estão relatadas na próxima seção.

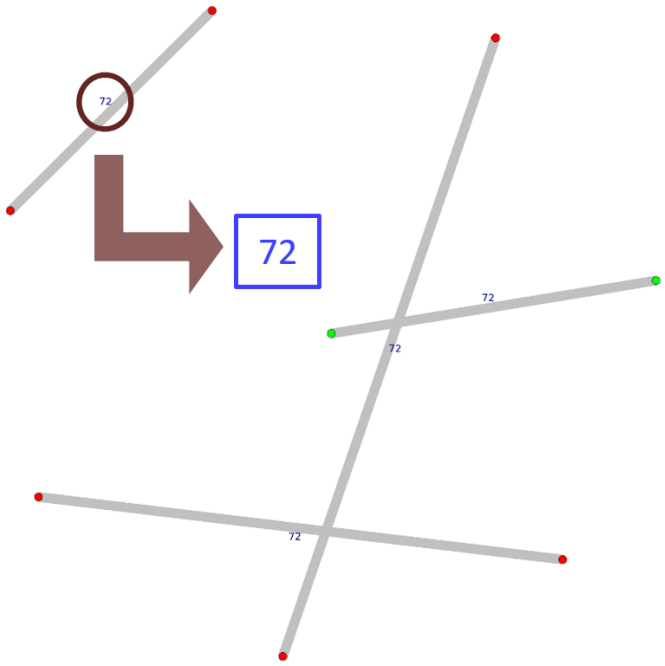

Figura 22: Grafo C3 com corte 100%

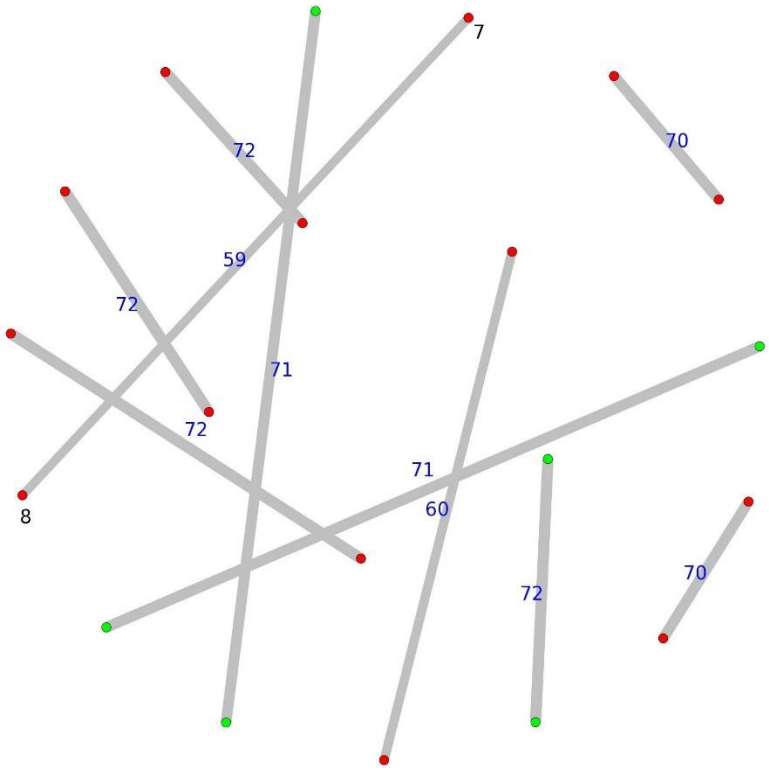

Figura 23: Representação em grafo da combinação C3 com corte de 80%

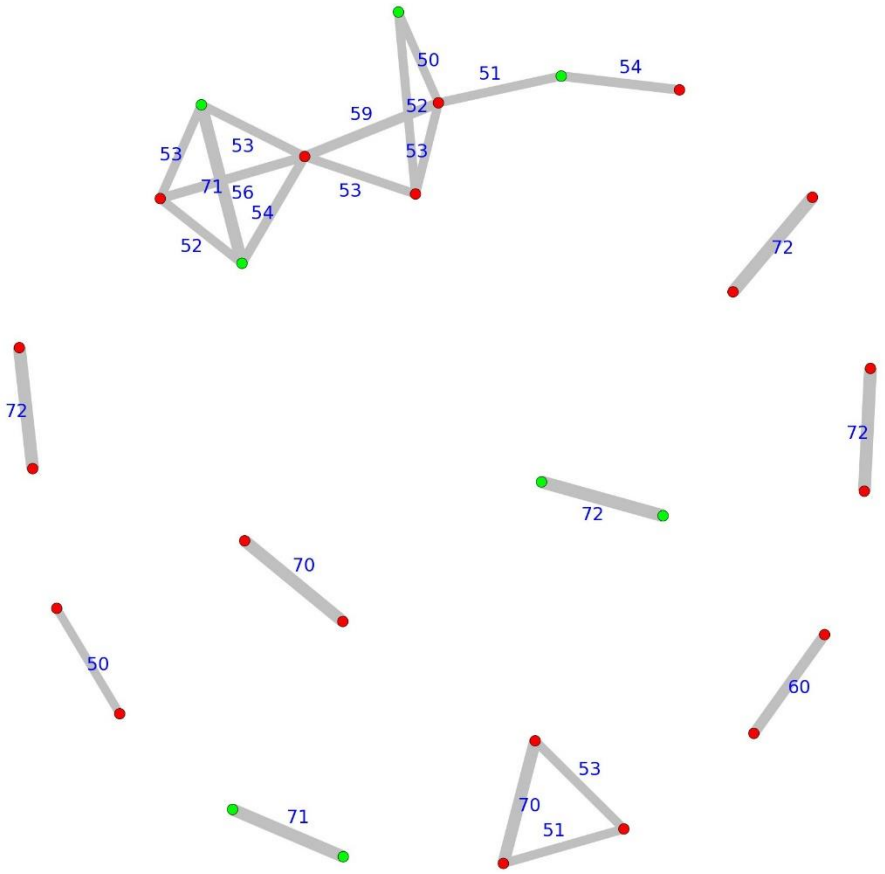

Figura 24: Representação em grafo da combinação C3 com corte de 70%

Capítulo 4

4.Análise de Resultados

A análise em grafos dos resultados do K-means através das maiores ligações proporcionou levantamentos bastante relevantes sobre os projetos de lei. Os resultados iniciais e alguns grafos representando as maiores ligações foram analisados pela especialista para que fosse identificado, através do conhecimento prévio sobre os projetos, se existia algum motivo claro para o alto nível de ligações entre eles. Foi constatado que os projetos presentes no corte 100% da combinação C3 são projetos equivalentes.

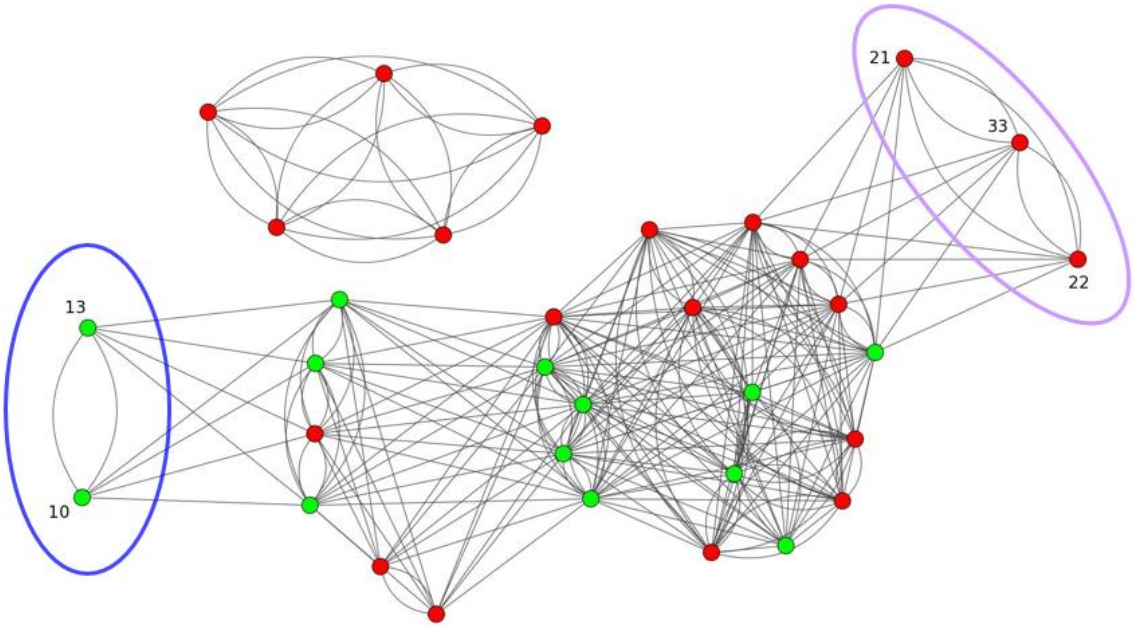

Figura 25: Representação em grafo da combinação C2 com destaque para alguns PLs

A Figura 25 representa o mesmo grafo da Figura 16, mas com destaque para algumas relações entre PLs que merecem atenção.

Como inicialmente relatado na seção 3.2.1, a partir da Figura 16 e com a ajuda da especialista, foram apresentados os projetos equivalentes. Para demonstrar este comportamento, foram destacados 2 grupos de PLs que possuem comportamentos semelhantes entre si: (10 e 13); (21, 22 e 33). Os pares (10 e 13) e (21 e 33), em especial, são projetos equivalentes entre si. O primeiro par (10 e 13) aborda como assunto comum

a criação de uma bolsa auxílio para mulheres que decidirem realizar aborto após estupro ou que sofreram aborto espontâneo. O segundo par (21 e 33), por sua vez, discorre sobre incluir um dispositivo na Lei nº 10.406, de 10 de janeiro de 2002 - Código Civil (Código Civil, 2017), dispondo sobre o registro público da gravidez. Estes pares, inclusive, são exemplos de PLs equivalentes totalmente idênticos, possuindo exatamente o mesmo texto. Essa alta equivalência entre estes PLs é verificada pelo algoritmo posteriormente, quando foi executada a combinação C3, e os pares apresentaram o número máximo de arestas entre eles (72). Assim, ambos os pares estão presentes também na combinação C3 com corte 100%, destacados na Figura 27, e consequentemente nos cortes menores como no 70% (Figura 24) onde cada par aparece separado dos demais. O PL 22, apesar de não ser um projeto equivalente aos outros (21 e 33), trata também a questão da obrigatoriedade do cadastramento da gestante, demonstrando assim que a análise também conseguiu identificar projetos que possuíam mesmo tema (assunto) e polaridade em textos diferentes (não equivalentes).

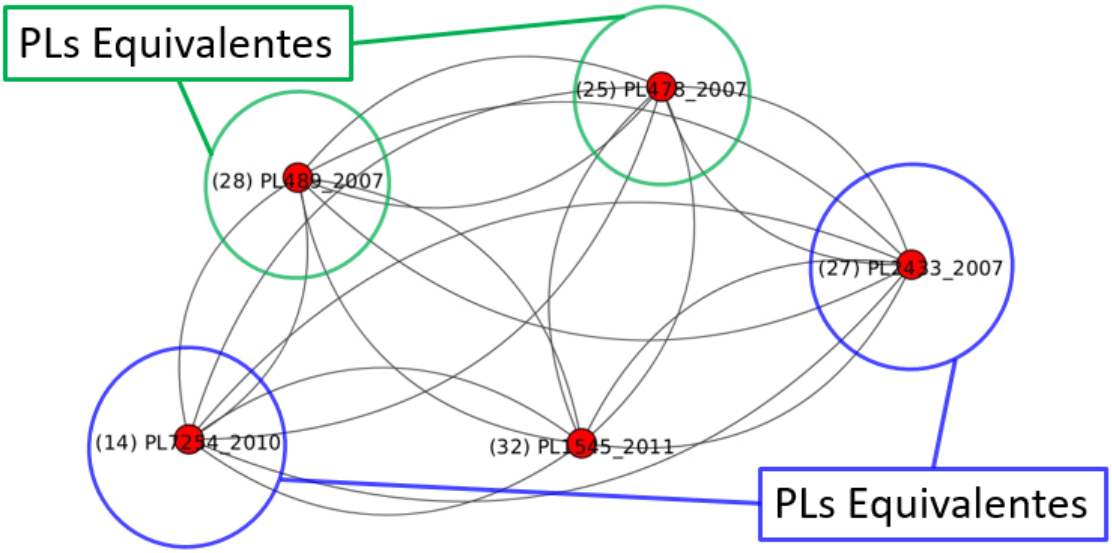

Figura 26: Destaque do grupo independente da Figura 25

Na Figura 26, como já introduzido na Figura 17, pode-se destacar que todos os PLs possuem viés negativo e só se ligam entre si, sendo que os dois mais acima ([28] PL478_2007 e [25] PL489_2007) correspondem ao projeto do estatuto do nascituro, que contém artigos similares do art. 2º do Código Civil, sendo considerados equivalentes. Ambos também estão representados como um par na combinação C3 com corte 100% (Figura 27) e, consequentemente, podem ser vistos nos cortes menores, como no corte de 70% (Figura 24) onde o par aparece separado dos outros, demonstrando a especificidade

do assunto. Os outros 3 projetos (14, 27, 32) possuem o mesmo objetivo, que é aumentar a pena de reclusão estipulada no Código Penal (2017) a terceiros que realizaram (ou provocaram) o aborto. Os PLs 14 e 27 são projetos equivalentes, apresentando o mesmo texto, com exceção de dois artigos adicionais no decreto (124 e 124-A), referentes ao aumento da pena de reclusão também às gestantes que provocaram ou consentiram o aborto e a tipificação do crime de induzir ou auxiliar a gestante à prática do aborto, respectivamente. Já o PL 32 concentra-se no aumento da pena para médicos, incluindo a proibição definitiva dos mesmos exercerem a profissão. A forte ligação entre estes três PLs (14, 27 e 32) é destacada (círculo) na Figura 28, onde, mesmo com um número bem maior de combinações, eles formam um triângulo isolado.

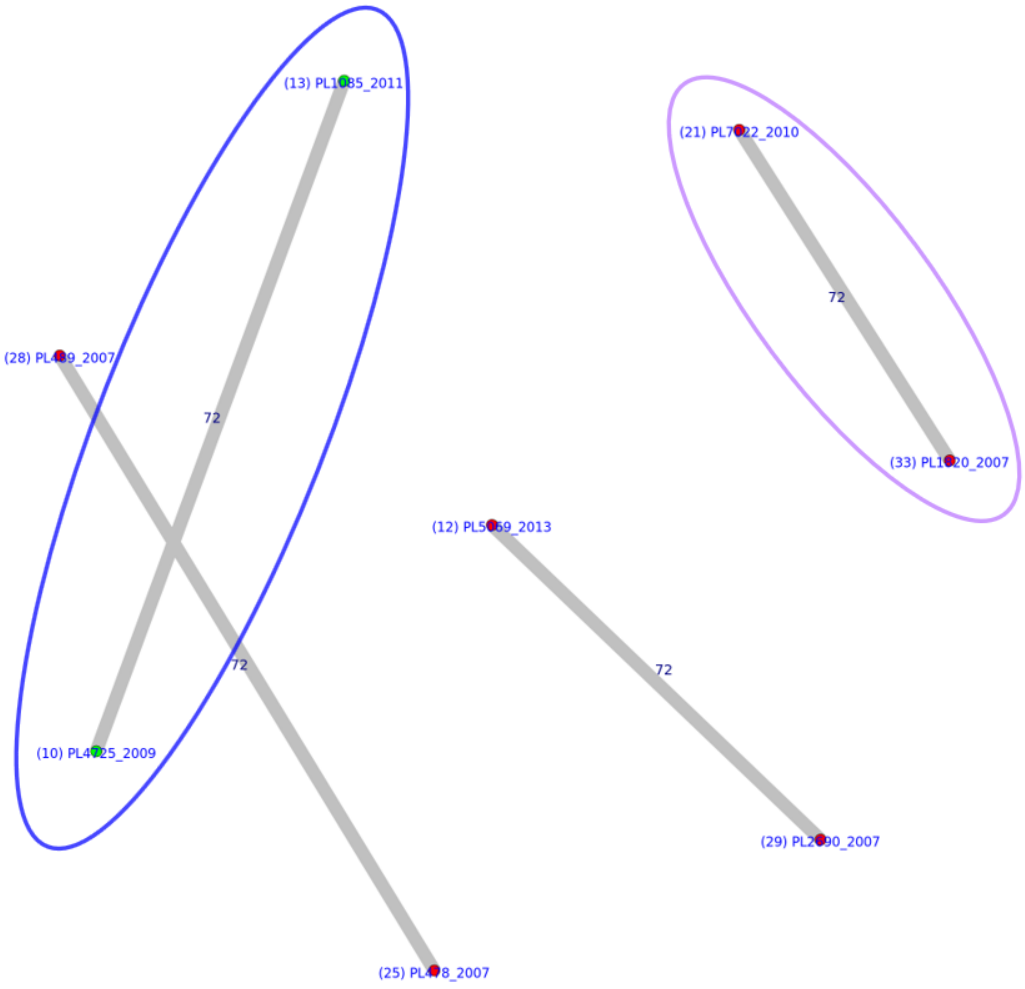

Figura 27: Grafo C3 com corte 100% com destaque para os dois pares

O único par ainda não analisado é o (12 e 29), que também se trata de um projeto equivalente relativo ao acréscimo ao Decreto-Lei no. 2.848 de 7 de dezembro de 1940 do artigo 127-A, que tipifica o crime de induzir, anunciar, provocar, orientar ou prestar

qualquer auxílio à gestante ao aborto. O PL 12 é uma reedição do PL 29, copiando praticamente todo o decreto, com exceção das penas que na reedição se tornam maiores, e grande parte da justificativa.

Todos os pares presentes no corte 100% são projetos equivalentes entre si, possuindo assim também a mesma polaridade. Esse comportamento se propaga até o corte 80% (Figura 29), onde todos os pares são projetos equivalentes e, consequentemente, de mesma polaridade, com exceção dos projetos 7 e 8, que tratam do aborto em casos de estupro, mas são propostas distintas: O primeiro suprime a possibilidade de aborto nesses casos, enquanto o segundo busca estabelecer uma barreira ao acesso ao aborto no SUS (com a obrigatoriedade do corpo de delito). Este é mais um exemplo onde o estudo conseguiu identificar projetos de mesmo tema (e mesma polaridade) ou objetivo, apesar de possuírem características e maneiras diferentes de abordagem. Este par foi outro caso de boa análise do algoritmo, que conseguiu identificar projetos com um mesmo assunto, mas com abordagens diferentes.

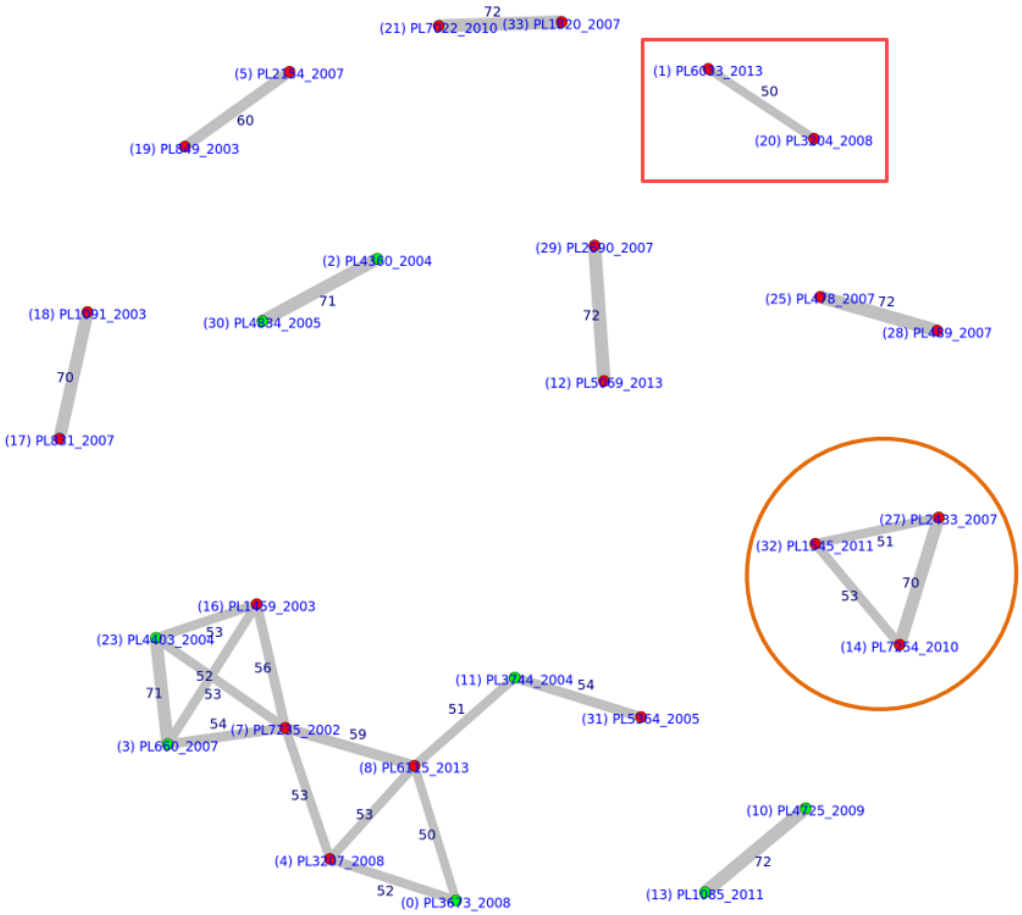

Figura 28: Combinação C3 com corte 70% - destaque para PLs

Com exceção dos pares (7 e 8), (14 e 27) e (3 e 23), todos os pares presentes no corte 80% continuam destacados no corte 70%, ou seja, não se ligam a outros PLs além do par. O par (14 e 27) já foi analisado e destacado na Figura 28 junto com o PL 32, assim como o par (7 e 8), destacado na Figura 29. O par (3 e 23) será analisado mais para frente junto com o subgrafo da Figura 30. Além disto, o único par destacado no corte 70% que não está presente no 80% é o par (20 e 1), destacado na Figura 28 por um retângulo, que inclusive apresentou o primeiro "erro" do algoritmo, visto que os dois projetos, apesar de estarem corretos quanto a polaridade, não apresentam semelhança aparente. O PL 20 discorre sobre a obrigação de constar uma advertência impressa nas embalagens de produtos comercializados para detecção de gravidez com os dizeres: "aborto é crime; aborto traz risco de morte à mãe; a pena por aborto provocado é de 1 a 3 anos de detenção". Já o PL 1 revoga a Lei 12.845 de 1º de agosto de 2013 que dispõe sobre o atendimento obrigatório e integral de pessoas em situação de violência sexual. Ambos apresentam decretos e justificativas muito curtas, em especial o PL 1 que só revoga a lei no decreto. Não foi possível identificar o porquê do alto número de ligações entre eles. Uma possível razão é o uso de um número elevado de termos iguais, ainda mais se considerar que ambos são textos pequenos, ou seja, com pouca informação.

O par (18 e 17), equivalentes, presente no corte 80%, discorre sobre a exigência para que hospitais públicos implantem um programa de orientação à gestante sobre os efeitos e métodos utilizados no aborto, quando este for autorizado legalmente, através de filmes, acompanhamento psicológico e etc. É mais um caso de reedição de projeto, no qual o PL 17 copia o decreto, e uma parte da justificativa do PL 18. Já o par (19 e 5), apesar de também serem equivalentes, possuem apenas trechos semelhantes na justificativa, mas uma redação diferente do decreto e justificativa no geral. Ambos têm por objetivo a criação de uma central de atendimento telefônico para recebimento de denúncias de abortos clandestinos, com o PL 5 frisando a reserva de um número telefônico de três dígitos para tal. O par (2 e 30), por sua vez, possuem uma justificativa idêntica, mas possuem leves diferenças no decreto, mas ainda sim são equivalentes. Ambos isentam de punição o aborto provocado por médico quando o feto é portador de anencefalia, comprovada por laudos independentes de dois médicos. O PL 2 também isenta de ilicitude a gestante que praticar o aborto nesta condição.

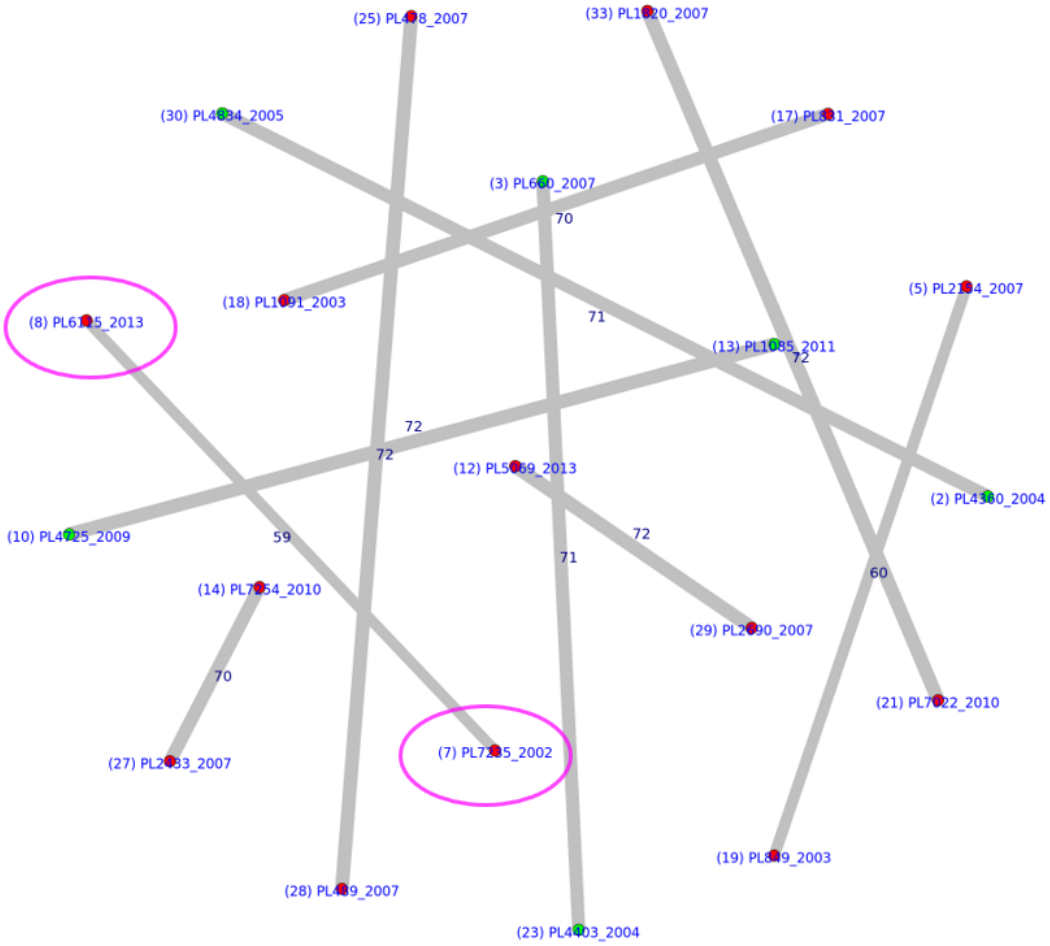

Figura 29: Combinação C3 com corte 80% - destaque para PLs 7 e 8

Um recorte com aproximação para melhor entendimento do subgrafo presente no corte de 70% está mostrado na Figura 30. A análise deste subgrafo é mais complexa, pois os PLs apresentam fortes ligações entre si, apesar de possuírem polaridades distintas em alguns casos. Por exemplo, os projetos 0 (PL3673_2008) e 4 (PL3207_2008) alteram a natureza do crime de aborto, mas com polaridades opostas: o primeiro o transforma em crime de “menor potencial ofensivo” e o outro o transforma em crime hediondo, além do induzimento, instigação ou auxílio ao suicídio. Já o projeto 8 (PL 6115_2013), como já levantado anteriormente, cria barreiras ao aborto em caso de estupro pelo SUS (com a obrigatoriedade do corpo de delito), com ligação maior para o projeto de mesma polaridade (PL 4). Outra análise interessante pode ser feita a partir dos PLs: 31 (PL5364_2005) e 11 (PL 3744_2004), pois os projetos possuem polaridades distintas. Ambos tratam da possibilidade de aborto praticado por médico em caso de estupro, porém, um expande essa possibilidade (incluindo atentado violento ao pudor), PL 11, e o outro elimina essa possibilidade e pune a prática. Este par é mais um caso de PLs com mesmo assunto, mas com polaridades opostas, demonstrando que o algoritmo conseguiu

identificar os assuntos, bastando apenas conseguir identificar a polaridade (possíveis soluções são levantadas nos trabalhos futuros, seção 5.1). Sobre a ligação entre os PLs 8 e 11, ambos dissertam sobre o aborto em caso de estupro, mas com polaridades opostas.

A relação entre os PLs 7 e 8 já foi discutida, assim como a relação entre 8 e 4.

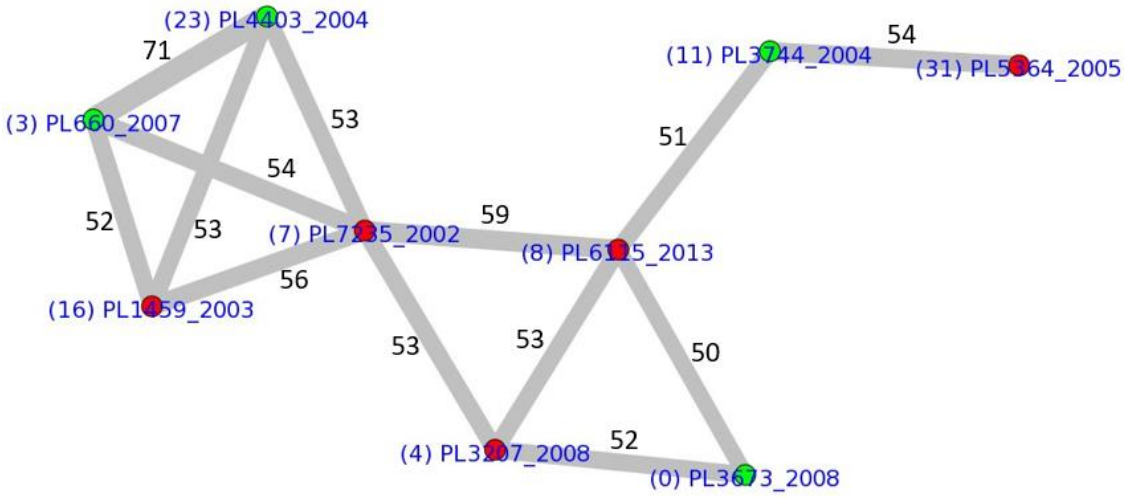

Figura 30: Destaque do grupo independente da Figura 28

Pode-se observar que a relação entre 7 e 8 é mais forte que as relações do PL 4 com cada um deles, o que, de acordo com o que já foi analisado sobre eles, faz sentido. O 4 se relaciona a natureza do crime de estupro, mas os PLs 7 e 8 apresentam uma proximidade maior, falando sobre aborto em caso de estupro.

Os projetos 3 (PL660_2007) e 23 (PL 4403_2004) são o mesmo projeto, projetos equivalentes, que propõem autorizar o chamado "aborto terapêutico", que consiste no aborto praticado por médicos em casos de evidência clinica embasada por técnica de diagnóstico complementar de que o feto apresenta anomalia grave ou incurável, incluindo anencefalia, que implique na impossibilidade de vida extrauterina. O projeto 3 é uma reapresentação do 23, contendo o mesmo decreto e uma justificativa atualizada, que inclusive cita o PL 23, mas que reutiliza grande parte do texto do projeto anterior. As 71 ligações entre eles confirmam essa equivalência. O projeto 16 (PL1459_2003) torna crime o aborto em caso de anomalia fetal, não definindo a gravidade da anomalia, ou seja, mesmo tema do par citado anteriormente (23 e 3), mas com diferente polaridade, punindo a prática nestes casos. Desta forma, a ligação entre estes três projetos é outro acerto do algoritmo, precisando apenas diferenciar a polaridade do PL 16 para os outros dois. Os PLs 16 e 7 possuem a mesma polaridade, ambos têm em comum a rigorosidade no trato em relação ao aborto, mas tratam de assuntos diferentes (aborto em caso de estupro e

aborto por anomalia fetal, respectivamente). Vê-se necessário identificar mais peculiaridades que propiciam as fortes ligações como esta que conseguem identificar polaridade e rigorosidade, apesar de não ser tão efetivo na detecção do assunto.

A Figura 31 apresenta um gráfico fornecido pela especialista com alguns projetos de lei e suas principais áreas que ajudou a visualizar e entender melhor o cenário de pesquisa.

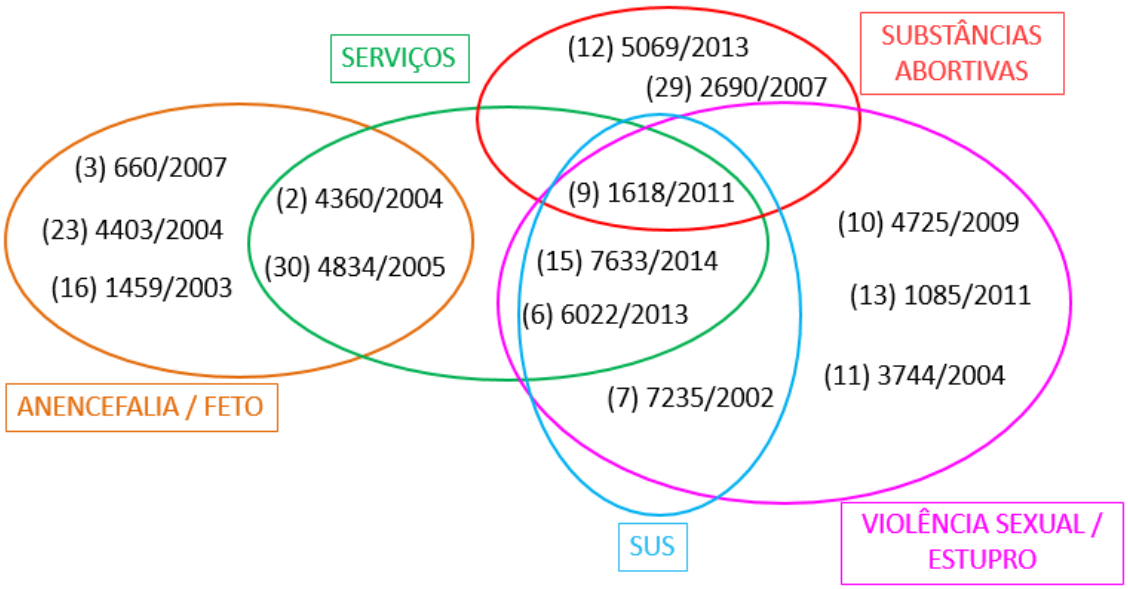

Figura 31: Diagrama de áreas de assunto de PLs

Capítulo 5

5.Conclusões

O estudo apresentado neste trabalho buscou o desenvolvimento de uma metodologia eficiente de identificação de polaridade de documentos de projetos de lei brasileiros. A metodologia apresentada se mostrou bastante eficiente no agrupamento de PLs. Em especial, foi possível identificar os "projetos equivalentes", que só passaram a ser conhecidos a partir da detecção do alto número de ligações entre eles e projetos similares no seu assunto, possuindo polaridades iguais ou opostas. Foi utilizado um algoritmo de clusterização que se mostrou bastante adequado ao propósito, e foi desenvolvida uma ferramenta eficiente de visualização dos resultados em grafos. Esta ferramenta foi elaborada devido às demandas de melhorias e simplificações do grafo para que fosse mais viável a sua análise manualmente. Todos os resultados e conclusões obtidas em relação aos PLs nesta dissertação se devem a todas as soluções relacionadas a representação em grafos.

A detecção de "projetos equivalentes" é muito útil pois automatiza o processo de análise, facilitando, por exemplo, o estudo de especialistas que poderão focar em projetos diferentes, não tendo que ler e analisar projetos equivalentes, ou seja que não possuem muita diferença e, caso as possuam serão mais facilmente detectadas, graças a um número menor de ligações, como, por exemplo, projetos equivalentes que só aparecem em corte abaixo de 100%, como foi o caso citado neste estudo do corte de 70% da combinação C3. Projetos semelhantes de mesma polaridade também foram encontrados pelo algoritmo, sendo mais um resultado útil e um dos objetivos iniciais deste estudo pois, junto com os projetos equivalentes, demonstram que esta técnica conseguiu polarizar corretamente estes documentos. Estes dois conjuntos podem, como descrito nos trabalhos futuros, ajudar na criação de uma metodologia que otimize o processo aqui definido, conseguindo desambiguar ligações não muito claras, como é o caso de clusters que possuem um grande número de elementos com polaridades distintas. Já os projetos semelhantes com polaridades distintas também foram resultados bastante satisfatórios, pois mostram que o algoritmo conseguiu identificar os assuntos de forma correta, embora não tenha conseguido detectar a polaridade. Estes casos também podem ajudar na melhoria da

metodologia, como descrito nos trabalhos futuros, modificando o pré-processamento para, por exemplo dar mais pontuação (TF-IDF) para certas palavras que identifiquem e diferenciem a polaridade desses documentos. Assim, eles propiciam uma identificação de padrões de similaridade entre PLs.

A identificação destes projetos equivalente e semelhantes também pode ser muito útil na automatização da apensação de projetos de lei na Câmara. Assim, projetos com assuntos semelhantes, tendo abordagens opostas ou não, poderão ser facilmente agrupados e apensados em um só.

Estes resultados podem, inclusive, ser favoráveis para a sociedade como um todo pois, com todas essas identificações de PLs, práticas que agrupem os documentos por assunto e polaridades iguais, por exemplo, podem facilitar o acesso das pessoas a esses documentos.

5.1 Trabalhos Futuros

Visa-se, posteriormente, utilizar toda a análise feita com os grafos e os PLs que possuem maiores ligações para identificar a polarização dos mesmos. Uma técnica levantada é a de se obter em cada grupo de polaridade, pré-classificado pela especialista, os principais N-grams, dando maior valor (TF-IDF) para os que mais aparecem nos grupos dos "top grafos" em destaque. Esta ação pode ajudar na desambiguação de grupos como o 3 da Figura 13, destacado novamente na Figura 32. Alguns estudos sobre esta técnica já foram feitos, como em Amorim (2011).

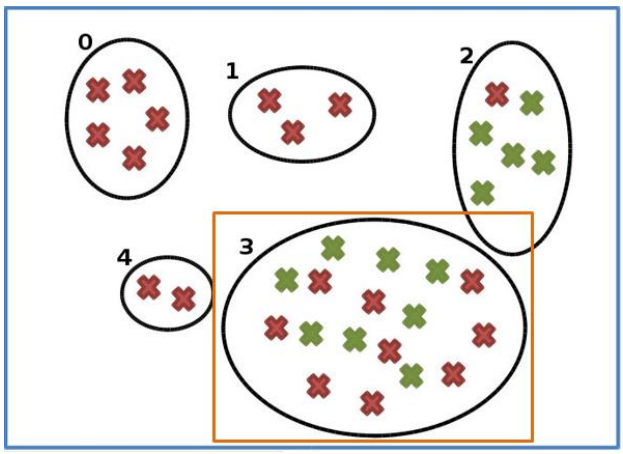

Figura 32: Combinação C1 com destaque para o grupo 3

Visa-se também utilizar métodos supervisionados, em especial o SVM, também nativo do scikit-learn, aplicando a visualização de sua classificação em grafos. Assim, pode-se comparar o resultado e eficiência dele com o K-means e identificar novas equivalências e semelhanças.

Outro trabalho futuro é o teste de ponderação de N-gram mais relevantes de polarização no método SVM, utilizando como base estudos anteriores como o de Joachims (1998) e Sista & Srinivasan (2004). Com isso, será possível comparar os resultados de ambos os métodos e analisar suas peculiaridades. Com isso, estudos citados anteriormente na Revisão bibliográfica seriam misturados à metodologia aqui levantada.

As listas com os principais N-grams identificados na seção 3.1 serão novamente analisadas após o agrupamento para poder identificar se estas aparecem em sua maioria em documentos contra ou a favor. Podendo assim ajudar na classificação, adicionando um processo hibrido de classificação não-supervisionada e supervisionada, ponderando estas palavras que tendem a ser usadas mais em um ponto de vista que de outro. Este método podendo ser testando de forma separada ou em conjunto com o método de detecção de palavras mais relevantes nos top grafos, definido no início desta seção. A especialista, analisando a lista de palavras mais frequentes gerada na seção 3.1, já conseguiu identificar algumas palavras que em sua maioria são usadas somente quando o projeto é de uma certa polaridade. Um exemplo disto é a palavra “nascituro”, característica de textos contra a liberalização do aborto.

Um dos problemas levantados na seção 3.1 é a existência de PLs com pouca informação, seja, por exemplo, com justificativas pequenas ou com decretos que somente revogam uma certa lei, não possuindo informação de sobre o que esta lei revogada trata. Uma solução que poderia contornar este problema é coletar os PLs apensados a este PL pouco informativo, utilizando o site da Câmara (Câmara, 2017), e identificar o assunto em comum destes apensados. No caso de decretos que somente citam uma lei ou artigo, outra solução é identificar e coletar o texto deles em sites do governo, como o site do Planalto (2017) que dispõe do texto do Código Penal (2017), e adicionar este texto no lugar da sua citação simples.

Por último, visa-se também testar a adição de termos à lista de *stopwords*. Estes termos a serem adicionados são ocorrências muito comuns no contexto do corpora e que não trazem muita informação nem ajudam a identificar a polaridade dos textos a que ele pertence.

Referências

Amorim, R., 2011. Learning Feature Weights for K-means Clustering Using the Minkowski Metric. PhD thesis, Birkbeck, University of London.

Apensados, 2017. Disponível em: <http://www.camara.gov.br/proposicoesWeb/fichadetramitacao?idProposicao=345103>. Acessado em: 11/11/2017

Arthur, D. & Vassilvitskii, S., 2007. K-means++: The advantages of careful seeding. In Gabow, H., eds, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '07), pp. 1027-1035.

Bird, S., Klein, E. & Loper, E., 2009. *Natural Language Processing with Python*. Disponível em: <http://www.nltk.org/book>. Acessado em 02/03/2017.

Câmara dos deputados. 2017. Disponível em: <http://www2.camara.leg.br/agencia/noticias/69896.html>. Acessado em: 03/07/2017

Código Civil, Lei nº 10.406. 2017. Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/2002/L10406.htm>. Acessado em: 03/07/2017

Código Penal, Decreto-Lei nº 2.848. 2017. Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto-lei/Del2848.htm>. Acessado em: 03/07/2017

Facebook, 2017. Disponível em: <https://www.facebook.com/>. Acessado em: 01/10/2017

Goyal, A. & Parulekar, A., 2015. *Sentiment Analysis for Movie Reviews*. Disponível em: https://pdfs.semanticscholar.org/4881/68a5403cd8804f37145f8a4bb78046a05be2.pdf. Acessado em: 20/10/2017

Harris, Z. S., 1954. Distributional Structure, *WORD – Journal of the international linguistic association*, vol. 10:2-3, p. 146-162.

igraph, versão 0.7.0 para Python. Disponível em: <http://igraph.org/python/>. Acessado em: 02/08/2017

Joachims, T., 1998. Text Categorization with Support Vector Machines: Learning With Many Relevant Features. In Nédellec, C. &Rouveirol, C., eds, *Machine Learning (ECML-98)*, p. 137-142.

Liu, B., 2012. *Sentiment Analysis and Opinion Mining.* Morgan & Claypool Publishers.

Liu, J., Shang, J., Wang, C., Ren, X. & Han, J., 2015. Mining Quality Phrases from Massive Text Corpora. In Sellis, T., Davidson, S. B. & Ives, Z., eds, *Proceedings of the*

2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15), p 1729-1744.

Liu, J, Shang, J. & Han, J., 2017. *Phrase Mining from Massive Text and Its Applications.* Morgan & Claypool Publishers.

Manning, C. D., Raghavan, P. & Schütze, H, 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Miner, G., Elder, J., Fast, A., Hill, T., Nisbet, R. & Delen, D., 2012. *Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications*. Elsevier.

Pang, B., Lee, L. & Vaithyanathan, S., 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP '02)* – Vol. 10, p. 79-86.

PL 5364/2005, 2017. Disponível em: <http://www.camara.gov.br/proposicoesWeb/fichadetramitacao?idProposicao=288053>. Acessado em: 05/10/2017

Planalto, Presidência da República, 2017. Disponível em: < http://www2.planalto.gov.br/>. Acessado em: 12/08/2017

Python, 2017. Disponível em: <https://www.python.org/>. Acessado em: 05/08/2017

Santos, D. L. B. & Silva, G. F. A., 2014. *Análise de Sentimentos em redes sociais – estudo de métodos aplicados em corpora na língua Portuguesa-BR do Facebook*. Trabalho de conclusão de curso Bacharelado em Ciência da Computação, Departamento de Ciência da Computação (DCC), Instituto de Matemática (IM), Universidade Federal do Rio de Janeiro (UFRJ).

scikit-learn. 2017. Disponível em: <http://scikit-learn.org>. Acessado em: 05/08/2017

Scipy, 2017. Disponível em: https://www.scipy.org/. Acessado em: 23/01/2017

Sista, S. P. & Srinivasan, S. H., 2004. Polarized Lexicon for Review Classification. In Arabnia, H. R. & Mun, Y., eds, *Proceedings of the International Conference on Artificial Intelligence (IC-AI '04)*, p. 867-872. CSREA.

SVM, Support Vector Machines, 2017. Disponível em: <http://scikit-learn.org/stable/modules/svm.html>. Acessado em: 07/06/2017

TfidfVectorizer, 2017. Disponível em: < http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>. Acessado em: 07/11/2017